

Turbonomic 8.9.3 Documentation



Chapter 1. Turbonomic 8.9.3 Documentation

Documentation for this product release includes the following general topics.

- [Release Notes](#) – Describes new features, improvements, deprecated/removed features, fixes, and known issues
- [Installation](#) – Helps you install Turbonomic in your on-prem environment
- [SaaS Setup](#) – Helps you set up a SaaS client in your on-prem environment, so that a SaaS server in the public cloud can manage your on-prem workloads
- [Product Overview](#) – Provides an overview of the platform and its underlying architecture
- [Getting Started](#) – Describes login steps, the Home Page, actions, and policies
- [Target Configuration](#) – Provides a list of targets that the product can monitor, and describes how to configure each target properly
- [User Interface Reference](#) – Provides a list entities discovered from targets, and describes how to configure plans, charts, and administrative settings
- [API Reference](#) – Helps you use the REST API as you script interactions with the product
- [Integration - Data Ingestion Framework](#) – Describes how to set up Data Ingestion Framework, a feature that enables you to define custom entities and entity metrics for your environment, and load them into the platform for analysis
- [Integration - Data Exporter](#) – Describes how to set up Data Exporter, a feature that streams Turbonomic data about entities and actions as JSON documents, and then publishes the data to a Kafka topic
- [Integration - Embedded Reporting](#) – Describes how to set up Embedded Reporting, an add-on that stores a history of your managed environment and presents this history via dashboards and reports

Turbonomic also integrates with ServiceNow, a third-party change management application. To view the documentation for this integration, visit this [page](#).

Chapter 2. Release Notes

Release Date: June 28, 2023

These release notes describe the following for Turbonomic 8.9.3:

- New features
- Feature updates and notices
- Improvements
- Fixes
- Known issues
- REST API changes

These release notes are for update to Turbonomic 8.9.3, from version 7.22.0 or later. For updates from the 7.21.x family or earlier, contact your Turbonomic representative.

To view the release notes for earlier versions of the product, go to this [page](#).

For any questions, contact [IBM Support](#).

What's New

The bi-weekly releases of Turbonomic include improvements to existing features, and new features that improve your experience with the platform. We have tested and documented these features, but they have not seen extensive use in existing customer environments. We invite you to try them and see how they improve the capabilities of the platform.

"What's New" is a running history of new features, as they have been added to different releases of Turbonomic.

Version 8.9.3

Note:

Frequent changes to the product or third-party targets require that some features are updated, deprecated, removed, or no longer supported. For details about these features, see [Feature Updates and Notices](#).

• Support for IBM® Power Systems

Initial feature support for managing IBM® Power Systems is now available. For IBM Power 8, 9, and 10 environments using Hardware Management Console (HMC) versions 9 or 10, Turbonomic will discover and continuously decide on the optimal logical partition (LPAR) processing unit allocation and virtual processor capacity based on historical demand collected from the HMC. IBM Power Systems and LPARs are represented as Hosts and Virtual Machines and are visible across all HMC targets in the Turbonomic UI. Recommended LPAR resize PU and VP actions will better inform IBM Power users on the optimal LPAR size. The recommended actions can leverage existing Turbonomic workflow capabilities for external approval and execution.

For details, see [IBM PowerVM](#).

Enable the PowerVM probe before adding the target. To enable the probe, set the following in the `charts_v1alpha1_xl_cr.yaml` resource:

```
spec:
  powervm:
    enabled: true
```

Note:

Features such as Virtual I/O Server (VIOS) LPAR optimization, move actions, planning, and full stack visibility are being considered for future releases. To submit Ideas for specific features that Turbonomic should consider, use the [IBM Ideas Portal](#).

- **Enhancements to Cloud Savings and Investments**

Turbonomic now uses cost snapshots and daily billing reports from your cloud providers to calculate top-down savings and investments associated with scale and delete actions. With this enhancement, savings and investments charts can now accurately report the impact of actions on your cloud expenses. In addition, the charts can now break down savings and investments by account, region, and other meaningful categories.

For details, see [Cumulative Savings and Investments Charts](#).

Note:

The charts will show savings and investments data for actions executed *after* the installation of version 8.9.3. Data for actions executed before this version will not be shown, but can be extracted from the platform using a script. If you need to extract the data, contact your Turbonomic representative for assistance.

- **IBM License Service in Turbonomic VM Images**

The IBM License Service is now included in Turbonomic Virtual Machine Images. The License Service collects license usage information that is used to create usage snapshots and reports. The License Service is installed automatically as part of the `t8cinstall.sh` script and gets deployed in its own namespace, `ibm-common-services`. After installation, the License Service user interface is available at https://<turbonomic_address>/licenseservice.

For more information, see [IBM License Service](#).

- **New Design Framework for the User Interface**

We are transitioning the product to a new design framework to deliver a high-performance user interface with a more modern look-and-feel.

In this release, the following features and pages use the new framework:

- License Management
- Maintenance
- Updates
- Email Settings

You can switch to the new framework via a toggle on the main menu. For details, see [Design Framework for the User Interface](#).

- **New View of the Target Configuration Page**

A new Target Configuration user experience is now enabled by default. This new view includes the following improvements:

- You can easily view targets in a tabular form with improved search, sort, and filtering capabilities.
- Targets with health issues are easily identified, alongside details such as who changed the target configuration and when it was changed.
- The Target Details page for some target integrations communicate the stages of validation and discovery, as well as any encountered error conditions to aid in troubleshooting your target configurations. Related targets are also shown where applicable.
- The target view no longer requires a manual refresh after adding or changing a target and is more dynamic than the previous view.

The option to use the older view is still available; however, this will be deprecated in a future release.

For details, see [Configuring Targets - Enhanced View](#).

- **Turbonomic SaaS Client Support for Kubeturbo Probes Running on a Cloud-Hosted Kubernetes Cluster**

Kubeturbo probes can now use the secure connector to interact with a Turbonomic SaaS instance and monitor Kubernetes clusters on Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS), and Google Kubernetes Engine (GKE). Kubeturbo probes that are deployed in a SaaS client environment automatically update to stay in sync with the Turbonomic SaaS version. The Turbonomic SaaS Client also collects logs from Kubeturbo probes.

For details, see [Installing the SaaS Client on Other Deployments](#).

- **Datadog Target Support for Containerized Entities**

The Datadog target now supports containers and related entities for Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS), and Google Kubernetes Engine (GKE), including discovery and stitching when the kubeturbo target is added.

- **RedHat Certified Ansible Hub Content Collection for Turbonomic**

The RedHat Certified Ansible Hub Content Collection for Turbonomic is now available to provide a starting point for integrating Turbonomic with Event Driven Ansible (EDA) using Webhooks. Turbonomic actions for scaling AWS or Azure instances can now be fed into EDA to trigger playbooks to make the changes required on AWS or Azure. The content collection, which includes a module to create the Turbonomic Webhook, roles, an example rulebook and playbooks can be found at these locations:

- [RedHat Ansible Hub Content Collection](#) (requires a Hub subscription)
- [RedHat Galaxy](#)
- [Public IBM GitHub repository](#)

Version 8.9.2

- **Sustainability Features for On-prem Environments**

Turbonomic now tracks the energy consumption and carbon footprint of on-prem hosts and VMs to help you put your sustainability goals into action.

This release introduces the following key features:

- Sustainability data is now available when you set the scope to hosts or VMs discovered via VMware vCenter, HPE OneView, and UCS Manager targets, and then view the **Energy** and **Carbon Footprint** charts.
- To calculate carbon footprint, Turbonomic uses industry standards that take into account energy consumption, data center efficiency, and carbon intensity data. You can create Data Center policies to adjust the calculations according to the requirements of your data centers. After you adjust the calculations, Turbonomic can accurately report your organization's carbon footprint.
- If you have enabled the Embedded Reporting add-on, use the sustainability dashboards to gain insight into how you can reduce your energy consumption and carbon footprint.

For details, see [Sustainability Features](#).

- **Parking Actions for AWS RDS Instances**

This release introduces 'parking' actions for AWS RDS instances to help you reduce your cloud expenses. With these actions, you can stop instances for a period of time and then start them when you need them. You can enforce parking actions on demand or according to a schedule.

Note:

AWS RDS instances are represented as Database Server entities in Turbonomic.

For details, see [Park: Stop or Start Cloud Resources](#).

Turbonomic requires the `rds:StopDBInstance` and `rds:StartDBInstance` permissions to stop and start instances. For the full list of permissions, see [AWS Permissions](#).

- **'Buy Reservations' Actions Based on Azure MCA Pricing**

Turbonomic can now recommend actions to buy Azure reservations based on MCA pricing, both for the real-time environment and for plans.

• Action Center Improvements

Action Center provides a comprehensive view of the actions that Turbonomic recommends to assure performance and improve infrastructure efficiency. To provide easy access to Action Center and the details for individual actions, we are introducing the following improvements:

- A new **Action** button is now available on the main menu of the user interface. This opens a dedicated Action Center page with a unique URL, which you can bookmark and share with interested parties.
- Action Center is now the default view for actions. Check out Action Center in entity views and plan results, and launch it conveniently from action-focused charts, such as the Pending Actions, Top Utilized, and Potential Savings charts.

For a list of Action Center features, see [Working with Action Center](#).

- The Action Details page for a specific action now has a unique URL (or deep link), so you can easily share the page with interested parties. To get the URL, click the 'share' icon at the top-right section of the page and then copy the URL displayed in the address bar of the new tab. For details, see [Action Details](#).

Version 8.9.1

• Support for Datadog Targets

This release introduces support for the Datadog target. This integration provides insights into application performance while optimizing resource allocation, ensuring that applications receive the necessary resources while minimizing waste, resulting in improved operational efficiency and cost savings. Turbonomic discovers the Business Application, Business Transaction, Service, Application Component, and Virtual Machine and its related commodities for Tomcat and Standalone Java applications.

For details, see [Datadog](#).

Enable the Datadog probe before adding the target. To enable the probe, set the following in the `charts_v1alpha1_xl_cr.yaml` resource:

```
spec:
  datadog:
    enabled: true
```

• Support for Azure Dedicated SQL Pools

Turbonomic now discovers [dedicated SQL pools](#) used in Azure Synapse Analytics, and represents them as Database entities in the supply chain. If there are no connections to a SQL pool for at least one hour, Turbonomic will recommend suspending the pool as a cost-saving measure.

For details, see [Database \(Cloud\)](#).

Turbonomic requires a new set of permissions to discover dedicated SQL pools and execute suspend actions. For details, see [Azure Service Principal and Subscription Permissions](#).

• Discovery of SQL Server Clusters

When you add SQL Server as a target, Turbonomic can now discover SQL Server clusters from the target, and represent the instances in those clusters as individual Database Server entities in the supply chain.

Turbonomic also creates a group for each cluster. When you set the scope to a group and click Database Server in the supply chain, the user interface displays a list of instances in the cluster, and indicates which instance is currently active and which ones are idle (redundant). Turbonomic only monitors resources for the active instance.

For details, see [SQL Server](#).

• Execution of VMware vCenter Rightsize Actions Together

If a VMware vCenter entity has several pending rightsize actions, they execute together when they are automated, manually accepted simultaneously, or approved before entering an execution schedule window. Executing rightsize actions together prevents multiple restarts of a single VM when there are

multiple pending actions, allowing for more changes within an execution window. This feature was enabled by default starting in version 8.8.6 and applies to all VMware vCenter rightsize actions.

Version 8.9.0

Platform Management

- **Turbonomic SaaS Client Deployment in Red Hat OpenShift Container Platform**

The Turbonomic SaaS client can now be deployed in the Red Hat OpenShift Container Platform. This deployment brings all of the capabilities of the Turbonomic SaaS client to the Red Hat OpenShift Container Platform, such as secure communication to SaaS and auto-upgrade of probes and the operator. It also provides a better SaaS experience with seamless access to new probe functionality and reduced maintenance of Turbonomic SaaS client deployments to Red Hat OpenShift Container Platform customers.

- **New Operator Lifecycle Management (OLM) Feature**

The new OLM feature automatically upgrades the Turbonomic client operator for customers utilizing the SaaS secure connector for hybrid support to provide a better SaaS experience with seamless access to new probe functionality, and reduced maintenance of SaaS client deployments.

Container Resource Management

- **Turbonomic SaaS Client Support for Kubeturbo**

On-premise Kubeturbo probes can now use the secure connector to interact with a Turbonomic SaaS instance. Kubeturbo probes that are deployed in a SaaS client environment automatically update to stay in sync with the Turbonomic SaaS version.

For details, see [cd](#).

- **Declarative Policy Configuration for Kubernetes Services**

This release introduces a declarative approach to creating and managing policies for horizontally scalable Kubernetes services. These policies adjust the number of pod replicas that back those Services to help you meet Service Level Objectives (SLOs) for your applications.

With this new approach, application owners who do not have access to the Turbonomic user interface can now use Custom Resource (CR) files in a Kubernetes cluster to create and manage policies. The settings in these policies are synced with Turbonomic every ten minutes to keep your environment up-to-date.

For details, see [Actions for Kubernetes Services](#).

- **Declarative Configuration of Prometurbo**

This release introduces a declarative approach to configuring Prometurbo as a collector of application performance metrics. Turbonomic uses these metrics to generate SLO-driven horizontal scaling actions for Kubernetes Services.

With this new approach, application owners can now use Custom Resources (CRs) to configure Prometurbo.

For details, see [Actions for Kubernetes Services](#).

- **Integration with Git-based Software via Argo CD**

Turbonomic now integrates with Git-based software via [Argo CD](#), a GitOps continuous delivery tool that manages ArgoCD Application resources using Custom Resources (CRs). With this integration, Turbonomic can identify the container cluster resources managed by Argo CD, and then execute container resize actions against the Git repository serving as Argo CD's 'source of truth'. Argo CD completes the action by synchronizing changes onto the running environment.

For details, see [Integration with Git-based Software](#).

- **Enforcement of Security Context Constraints (SCCs) for Pod Move Actions**

Red Hat OpenShift uses [SCCs](#) to control permissions for pods. When executing pod move actions, Turbonomic now carries over the user-level SCCs of a pod to its new node. This prevents privilege escalation issues that occur when pods run with admin-level SCCs in their new nodes.

For details, see [Pod Move Actions](#).

- **Temporary Increases in Namespace Quotas**

When a namespace has a defined quota and a Workload Controller in the namespace requires a resize, Turbonomic now increases the quota temporarily to ensure that the execution of the resize action complies with your [rolling updates](#) strategy.

For details, see [Namespace Actions](#).

- **Downloads of Resize Data for Workload Controllers**

You can now download data for resize actions executed by Workload Controllers to gain insight into workloads with performance risks due to resource limits, as well as opportunities to reclaim unused resource requests.

For details, see [Workload Controller](#).

On-prem Resource Management

- **VMware vCenter 8.0 Support**

This release introduces support for VMware vCenter version 8.0. To submit Ideas for specific vCenter 8.0 features that Turbonomic should consider, use the [IBM Ideas Portal](#).

Cloud Resource Management

- **Parking Actions for Cloud VMs**

This release introduces 'parking' actions for VMs in the public cloud to help you reduce your cloud expenses. With these actions, you can stop VMs for a period of time and then start them when you need them. You can enforce parking actions on demand or according to a schedule.

For details, see [Park: Stop or Start Cloud Resources](#).

- **Support for Microsoft Customer Agreement Accounts**

The Azure Billing target now supports [Microsoft Customer Agreement](#) accounts. When you add an Azure Billing target and specify the Billing Account ID for an account, Turbonomic discovers pricing for the workloads billed under that account, and then uses pricing information when recommending actions for workloads.

For details, see [Azure Billing](#).

- **Support for the Cost Details API**

The Azure Billing target can now collect billed cost data asynchronously via the [Cost Details API](#). This API is the default data collection method starting in this release, and is intended to eliminate the need to configure a cost export and its associated storage and container permissions.

If you set up a cost export previously, the Azure Billing target will continue to use that cost export. You can switch to the API if your billed cost data is less than 2 GB in size.

For details, see [Azure Billing](#).

- **Azure VM Memory Metrics Collection via REST API**

Turbonomic now supports collecting host OS based (agentless) memory metrics for Azure VMs, based on the REST API (Preview) recently made available by Azure. This mechanism is used as a fallback to collect VM memory when other agent-based VM memory collection mechanisms are not configured. Azure Memory Source Groups have been updated to better reflect the supported VM memory sources.

For details, see [Azure Memory Metrics Collection](#).

- **Billed Costs in Cloud Charts**

We have enhanced the way we collect billed costs from your cloud provider to give you a more accurate view of cost data in charts. Data in these charts should now closely match the costs you see in your billing reports.

To view billed cost data, add the Top Accounts, Cost Breakdown by Tag, and Expenses charts to your dashboards. You can customize the Expenses chart to display Billed Cost by Service Provider, Top Billed Cost by Account, Top Billed Cost by Service, or Workload Cost Breakdown.

Note:

The Billing Breakdown and Estimated Cost Breakdown charts will be removed from the product in an upcoming release. If you have added these charts to your dashboards, delete them immediately and start using the replacement chart, Workload Cost Breakdown.

- **Discovery of Account Tags from Public Cloud Providers**

Turbonomic now discovers tags associated with accounts across all supported public cloud providers. You can use these tags when you search for accounts or create groups of accounts.

Application Performance Management

- **Collection of Database Metrics from Instana**

Turbonomic can now collect Oracle and MySQL database metrics from Instana targets. Metrics include cache hit rate, database memory, database connections, and transaction logs.

User Interface Management

- **New View of the Target Configuration Page**

This release introduces a new view of the Target Configuration page to help you manage targets with ease. This new view includes the following improvements:

- The page lists your targets in a sortable table and provides a new target editing experience.
- When you edit a target, links to the [IBM Turbonomic Community](#) and [IBM Turbonomic Support](#) sites are available to quickly provide any assistance you may need.
- For some targets, the page shows stages of validation and discovery, as well as any errors to help with troubleshooting.

Note:

Search and filter features will be introduced in the future.

Try the new view by navigating to **Settings > Target Configuration** and turning on **New View** at the top-right section of the page. To provide feedback, you can enable telemetry in **Settings > Maintenance Options**. To submit Ideas to improve the target experience, use the [IBM Ideas Portal](#).

For details, see [Configuring Targets](#).

- **Component Health Information**

For Turbonomic instances deployed in on-prem environments, the Notification Center now includes Component Health information to notify you of product component issues and the steps you can take to resolve these issues.

To open Notification Center, click the information icon on the main menu.

Feature Updates and Notices

Frequent changes to the product or third-party targets require that some features are updated, deprecated, removed, or no longer supported. See the following sections for more information about these features.

Updates

The updates listed here will be made in the noted version. Consider the details and recommended actions that are provided.

Feature	Status	Details and Recommended Action
Azure permissions	Updated in version 8.9.1	Permissions were updated to more accurately reflect the minimum permissions for discovering workloads and executing actions. For example, permissions with wildcards (*) were replaced with the exact permissions. For details, see Azure Service Principal and Subscription Permissions .
Rate of Resize setting in on-prem VM policies	Updated in version 8.9.1	The Rate of Resize default value has changed from 2 to 3. If you have changed your default setting to 1 or want to keep the current default setting of 2, create a new policy scoped to all on-prem VMs and configure the Rate of Resize to your desired setting.

Notices

Features are labeled based on the following definitions:

- **Deprecated** – The feature is still supported but no longer developed or enhanced. The feature is not recommended for use and might become obsolete. IBM might remove it in a subsequent release of the product.
- **Removed** – The feature is no longer available in the product.
- **Unsupported** – The feature is no longer supported in the product.

Note:

When a specific release or version of an integration partner technology reaches end-of-life (EOL) or its end of support date, Turbonomic no longer provides support for that version. Turbonomic follows integration partners' official EOL timeline for version support. Targeting a non-supported version, or one that is no longer supported by the vendor, is at your own risk.

The changes listed here will be made in the noted version. Consider the details and recommended actions that are provided.

Feature	Status	Details and Recommended Action
MySQL 5.7	To be unsupported in a future release	MySQL version 5.7 will reach EOL on October 21, 2023 and will no longer be supported after that date. Turbonomic plans to support MySQL 8.0 in a future release as an optional alternative database server for MariaDB. MySQL version 5.6 reached EOL on February 5, 2021. Because it is no longer supported, references to MySQL 5.6 were removed from the documentation in version 8.9.0. For details, see EOL MySQL .
Billing Breakdown and Estimated Cost Breakdown charts	Removed in version 8.9.1	The charts have been removed from the product and can no longer be added to dashboards. If you have added these charts to your dashboards, delete them immediately and start using the replacement chart, Workload Cost Breakdown.

Feature	Status	Details and Recommended Action
Microsoft Enterprise Agreement target	Deprecated in version 8.8.6	Deprecation impacts customers who previously added, or plan to add, a Microsoft Enterprise Agreement target to manage Azure non-government subscriptions. For details, see Microsoft Enterprise Agreement .
VMware vCenter versions 6.0, 6.5, and 6.7	Unsupported since version 8.8.3	Update your VMware vCenter version to 7.0 or 8.0, which are fully supported.
Tbmigrate, the Classic-To-XL Migration Tool	Unsupported since version 8.5.0	If you need to migrate from a Classic installation to one of the 8.x version families, contact your support representative.

Versioning Explanation

Turbonomic versioning uses V-R-M elements (Version, Release, Modification) in the version number to express the status of a release.

Numbered Element	Example	Description
V - Version number	8.X.X	<ul style="list-style-type: none"> Changes to platform architecture or significant changes to data models
R - Release number	X. 2 .X	<ul style="list-style-type: none"> Major feature changes
M - Modification number is 0 (zero)	X.X. 0	<ul style="list-style-type: none"> A quarterly release All Early Access (EA) features from previous bi-weekly releases are now GA No new Early Access (EA) features in this release
M - Modification number is greater than zero (1 or higher)	X.X. 3	<ul style="list-style-type: none"> A bi-weekly release Can include new Early Access (EA) features Includes fixed issues <p>Note: For API developers, the X.X.1 release can include final implementations of deprecated API features. These final implementations can make API changes that are not compatible with an earlier version.</p>

Configuration Requirements

For this release of Turbonomic, you must satisfy the following configuration requirements.

Dynatrace Targets

Starting with Turbonomic version 8.6.2, the API token that you use when you configure a Dynatrace target must access specific scopes of the Dynatrace API V1 and V2.

If you are updating to Turbonomic version 8.6.2 or later, from a version that is earlier than 8.6.2, you must generate a new API token for the Dynatrace target configuration. Then, you must enter that token in the target configuration, and validate the target.

Turbonomic uses the API token to authenticate its calls to the Dynatrace API. This token must have permission to run GET methods using the Dynatrace API, both Version 1 and Version 2. Generate a new generic access token with these scopes:

Turbonomic Functions	Required Permissions
Monitoring	<ul style="list-style-type: none"> • API V1 scopes <ul style="list-style-type: none"> – Access problem and event feed, metrics, and topology • API V2 scopes <ul style="list-style-type: none"> – Read entities – Read metrics

Note:

If the target still fails to validate after you update the access token, take note of your configuration settings, delete the target, and configure the target again. Be sure to use the new API token that you generated.

Turbonomic Updates and Operator Version

Turbonomic deploys as a cloud-native application on a Kubernetes cluster. This cluster can be pre-configured on a VM that you deploy, or you can deploy Turbonomic to a Kubernetes cluster in your environment. In either case, Turbonomic uses an Operator to manage the application deployment.

For different versions of Turbonomic, the version of Operator you use changes as follows:

Product Version	Operator Version
8.9.3	42.34
8.9.2	42.33
8.9.1	42.32
8.9.0	42.31
8.8.6	42.30
8.8.5	42.29
8.8.4	42.28
8.8.3	42.27
8.8.2	42.25
8.8.1	42.24
8.7.6 - 8.8.0	42.23
8.7.5	42.22

Product Version	Operator Version
8.7.4	42.21
8.7.3	42.20
8.7.2	42.19
8.7.1	42.18
8.6.6 - 8.7.0	42.17
8.6.4 - 8.6.5	42.16
8.6.3	42.15
8.6.2	42.14
8.6.1	42.13
8.6.0	42.12

When you update Turbonomic, always include the matching version of Operator in the update. Online or offline updates that were completed according to the [latest installation instructions](#) automatically include the latest Operator.

If you installed Turbonomic on a Kubernetes cluster, you might need to manually update the Operator version.

Red Hat OpenShift Installations of Turbonomic

If you manage your installation with the Red Hat OpenShift OperatorHub, see [Installing using the Red Hat OpenShift OperatorHub](#).

Other Kubernetes Installations of Turbonomic

Note:

Starting with Turbonomic version 8.7.5, IBM Container Registry is used for all Turbonomic images for online upgrades and new installs. Ensure you have access to <https://icr.io>.

For installations on other supported Kubernetes platforms, you can update the Operator version in either one of two ways:

- Directly edit the running deployment of Turbonomic:
 1. Enter Edit mode for your cluster.


```
kubectl edit deployment t8c-operator -n {YourNamespace}
```
 2. Edit the Operator image.

Find and edit image: `icr.io/cpopen/t8c-operator:{Operator version}`
 3. Edit the registry, if not already set to use IBM Container Registry.

Find and edit registry: `icr.io/cpopen/turbonomic`
 4. Validate that the Operator pod is ready.

Run the command: `kubectl get pods -n {YourNamespace}`, and verify that the pod is running and ready.
- Edit the Turbonomic deployment YAML file:
 1. Open the Operator deployment file for editing.

In the location where you store your manifests, open the file `operator.yaml`, which is the file that you use to deploy the `t8c-operator` pod.

2. Edit the Operator image.

Find and edit `image: icr.io/cpopen/t8c-operator:{Operator version}`.

3. Edit the registry, if not already set to use IBM Container Registry.

Find and edit `registry: icr.io/cpopen/turbonomic`.

4. Apply the change to the operator.

Run the command: `kubectl apply -f operator.yaml`.

5. Validate that the Operator pod is ready.

Run the command: `kubectl get pods -n {YourNamespace}`, and verify that the pod is running and ready.

After you update the Operator version, and you verify that the pod is running and ready, edit your Custom Resource declaration to update Turbonomic to the version that matches your Operator version.

For more information, contact your support representative.

Supported MariaDB Version for OVA and VHD Installations

For its default historical database on OVA and VHD installations, Turbonomic currently supports MariaDB version 10.5.20. This support includes comprehensive testing and quality control for Turbonomic usage of the historical database.

Important:

Because of a known issue, *you must never use* MariaDB versions 10.5.14, 10.5.15, 10.6.7, 10.7.3, or 10.8.2.

If you are running Turbonomic installed as an OVA or VHD image, and are using the database that is included in that installation, then you must use version 10.5.20. For versions of Turbonomic that you installed as an OVA or VHD before version 8.7.6, you must now update to MariaDB version 10.5.20 if not already done.

For more information, see [Verifying Your MariaDB Version](#).

Required Database Capacities

For Turbonomic versions 8.0.6 or later, your historical database must provide certain storage size capacities. For MariaDB or MySQL installations, you must ensure that your database provides the necessary messaging and logging capacity.

If you installed Turbonomic as an OVA, and use the included MariaDB for the historic database, you can set the correct capacities by updating your Turbonomic deployment to the latest version. For more information, see [Increasing Your Database Capacities](#).

SQL Server Modes for External Databases

If you deploy Turbonomic to work with an external database instead of the included historical database, then you must specify the correct SQL Server modes for the database. Configure the database as `{{ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION}}`.

In particular, the SQL Server modes should not include `ONLY_FULL_GROUP_BY`, `NO_ZERO_IN_DATE`, or `NO_ZERO_DATE`.

External Databases and Turbonomic Updates

If you deployed Turbonomic with an external database server, for some updates you might need to manually create a new database and user for that deployment.

Note:

If your external database server is multi-tenant, or if your database server does not grant administrative privileges to Turbonomic, then you must continue with this configuration requirement.

Azure database services are multi-tenant. If you deployed an external database on Azure, this configuration requirement applies to you.

If you deployed your database server in a way that grants Turbonomic privileges to create new databases and new users, then the update automatically creates the necessary database. This configuration requirement does not apply to you.

For some Turbonomic updates, the updated version includes new databases on the server. If you are updating to one of these versions, then you must first create the new database and a user account with privileges to access that database. Then, you can update to the latest version of Turbonomic.

For more information, see [External DBs and Updates](#).

Transport Layer Security Requirements

By default, Turbonomic requires Transport Layer Security (TLS) version 1.2 to establish secure communications with targets. Most targets have TLS 1.2 enabled; however, some targets do not enable TLS or they enabled an earlier version. In that case, you see handshake errors when Turbonomic tries to connect with the target service. When you go to the Target Configuration view, you see a Validation Failed status for such targets.

NetApp filers often do not enable TLS and the supported version is TLS 1.0, which causes the NetApp target to fail validation.

If target validation fails because of TLS support, you see validation errors with the following strings:

- No appropriate protocol

Ensure that you enable the latest version of TLS that your target technology supports. If you still encounter issues, contact IBM Technical Support.

- Certificates do not conform to algorithm constraints

Refer to the documentation for your target technology (such as the NetApp documentation) for instructions to generate a certification key with a length of 1024 or greater on your target server. If you still encounter issues, contact IBM Technical Support.

Improvements

- **Improvement:**

The `Server Configuration` filter is now available.

For SQL Database Servers discovered from SQL or APM targets, the `Server Configuration` filter is now available when you search for servers or create server groups. This filter returns a list of standalone or clustered servers.

- **Note:**

The filter supports two options: clustered and standalone. However, currently only MSSQL (Microsoft SQL Server) databases support both scenarios. For all non-MSSQL databases, including database servers like Oracle and MySQL, we recommend using the filter in the standalone mode by default. This means Turbonomic will recognize and treat the database as a standalone configuration.

Known Issues

- **Known Issue:**

The Top VM and Top Host widgets on the PowerVM dashboard may display incorrectly.

The Top VM and Top Host widgets on the PowerVM dashboard might encounter an issue related to temporary groups, which are ad-hoc groups created in the following scenarios:

- When you click on the nodes or donut in the supply chain
- When you select entities in the search and scope the selection

To work around this issue, you can do one of the following:

- Edit the Top VM or Top Host widget to add the power commodities.
- Reopen the PowerVM group.
- Edit the URL to remove the `?entityType=` query string parameter.

- **Known Issue:**

For vCenter Server environments, Turbonomic does not recognize DRS rules for VM restart dependencies that are based on VM-VM Dependency rules.

For vCenter Server environments, Turbonomic does not recognize DRS rules for VM restart dependencies that are based on VM-VM Dependency rules.

You may be able to achieve a similar effect by expressing dependencies via the VM-Host rule, or cluster affinity or anti-affinity rules.

- **Known Issue:**

For AppDynamics environments, the platform cannot discover Database Servers if the target authentication uses oAuth for credentials.

For AppDynamics environments, Turbonomic cannot discover Database Servers if the target authentication uses oAuth for credentials.

- **Known Issue:**

Changes to a policy do not immediately show up in the user interface view of the affected scope.

When you set the scope of the Turbonomic view to a group, you can then view the automation policies that impact the given group. If you edit a policy for that group (in Settings: Policies), and then scope the view to that group again, the policy changes do not appear in the display for that group.

The display should update within ten minutes, after the next round of incremental discovery. If the condition persists, log out of your session and log in again to update the display.

- **Known Issue:**

Optimized Improvements for plans do not include hosts to provision.

For cases where actions indicate provisioning new hosts, the Optimized Improvements chart does not include the hosts to provision in the After Plan section.

- **Known Issue:**

The Optimal Improvements chart can show incorrect data for hosts to be suspended.

In cases where actions recommend that you suspend hosts, the Optimal Improvements chart should indicate no utilization on the hosts to be suspended. Under some circumstances, the chart can show utilization on these hosts. The result is incorrectly low values for utilization on the other hosts in the current scope.

- **Known Issue:**

For Turbonomic deployments where the database uses a custom (non-3306) port for MariaDB or MySQL, the Parking feature cannot start.

For Turbonomic deployments where the database uses a custom (non-3306) port for MariaDB or MySQL, the Parking feature cannot start. In these cases, suspend should be disabled from the `charts_v1alpha1_x1_cr.yaml` resource as follows:

```
spec:
```

```
suspend:  
  enabled: false
```

- **Known Issue:**

ThoughtSpot Liveboard queries that take more than 10 minutes to complete result in a query timeout message.

ThoughtSpot Liveboard queries that take more than 10 minutes to complete result in a query timeout message.

- **Known Issue:**

New Relic has ended support for monitoring integration with Microsoft SQL Server 2012.

Turbonomic no longer supports monitoring and stitching of Microsoft SQL 2012 discovered through New Relic. We recommend that you upgrade your Microsoft SQL instance to a version New Relic supports.

- **Known Issue:**

For Azure, billing information for RIs can show costs as NA.

In Azure environments, if you configure a Microsoft Enterprise Agreement target, costs for RIs can show in the user interface as NA. For example, a VM that shows 100% RI coverage can show the Reserved Compute Cost as NA. This occurs because of a known gap in the data provided by the Microsoft EA API that the target probe uses to collect data.

- **Known Issue:**

Cloud VMs might scale to very large instance types even if there are smaller instance types available for scaling.

When a cloud VM with a specific disk count and disk type applies a policy that enables 'instance store aware scaling', Turbonomic might recommend scaling the VM to a very large instance type, even if there are smaller, less expensive instance types that can adequately meet the VM's resource requirements.

To avoid this issue, disable instance store aware scaling and restrict the VM to its current instance family. For example, if an AWS VM is currently running the `i3.2xlarge` instance type, specify the `i3` instance family as a scaling constraint for the VM.

- **Known Issue:**

For New Relic MySQL, DB Cache Hit Rate values are incorrect.

For New Relic MySQL 8, DB Cache Hit Rate values are incorrect in Turbonomic. Because the percentage of queries that are retrieved from the cache (`db.qCacheHitRatio`) is not supported in New Relic MySQL version 8.0 and higher, the DB Cache Hit Rate values are no longer displayed in Turbonomic.

For more information, see the [New Relic documentation](#).

- **Known Issue:**

The user interface shows incomplete Azure billed costs and metrics because of an issue with the Azure API.

When attempting to discover Azure billing targets configured with partitioned cost exports on the first day of any month, Turbonomic is unable to find the directory containing the export files within a customer's Azure Storage account. The export files exist, but the Azure API does not return the correct storage location of that day's cost export files. As a result, the Turbonomic user interface may not reflect any billed cost that Azure exported on the first of the month, and the metrics dependent on that billed cost. The billed cost, including costs for the first day of the month, will start appearing on the second day of the month. Note that the billed cost is not limited to expenses incurred on the first day of the month.

- **Known Issue:**

Customer Issue: 120842

In a Nutanix environment, a Replace Hosts plan can fail to place VMs.

You can configure a Replace Hosts plan on a Nutanix cluster to replace the hosts with HCI templates. However, the plan will fail to create the HCI hosts, and will result in unplaced VMs.

- **Known Issue:**

Hardware Refresh to replace hosts with HCI templates can fail to place workloads.

When running a Hardware Replace plan, the plan can fail to place workloads onto HCI hosts. The plan correctly places workloads if the plan scope is in a hyperconverged environment. If the scope is *not* in a hyperconverged environment, then you must scope the plan to an entire cluster, and you must configure the plan to replace all the hosts in the cluster with HCI templates.

- **Known Issue:**

For Kubernetes, in some environments analysis cannot execute Scale Node actions.

For Kubernetes OCP 4.x and AKS environments, Turbonomic can generate and execute Scale Node actions. However, if the environment includes other Kubernetes distributions that don't support execution of Scale Node actions (EKS, AKS, and OCP), then Turbonomic can disable execution of *all* Scale Node actions in the environment.

- **Known Issue:**

For Kubernetes environments with nodes running Linux with cgroup v2 enabled, you must use Kubernetes version 1.23.2 or later.

For Kubernetes environments with nodes running Linux with cgroup v2 enabled, an issue in earlier versions of Kubernetes prevents Turbonomic from collecting CPU utilization data for the affected nodes. To collect CPU utilization from the cgroup v2 nodes, you must run Kubernetes version 1.23.2 or later.

- **Known Issue:**

Customer Issue: 114942

With Executed Actions charts, some data is missing for actions on entities that have been removed from the environment.

When you view Executed Actions charts or export data from them, some data is missing for actions on entities that have been removed from the environment. For example, assume an action was executed on a storage volume, and that volume has later been removed from the environment. In that case, the exported data for that action will not include values that describe the removed volume.

REST API Notes

This section describes issues that result in changes to the REST API. It is a running list of changes that accumulates until the next release of the API Guide. When we release a new version of the guide, it will include these changes. At that time, this list will start with new changes to the API.

Because the API is a full reflection of the Turbonomic product, some changes to the product necessarily cause changes in the API. This can include bug fixes and improvements to the product. For example, we might add new commodities that you can get for a given entity type, or a bug fix might change the data that is returned by an API request. We strive to make these changes backward-compatible whenever possible.

In some instances, we must make changes that are not backward-compatible. In this case, we deprecate the change for a period of time that should be sufficient for you to understand the change and implement an alternative in your scripts.

REST API Changes

- **Improvement:**

Fix Version: 8.9.1

The Parking APIs are now available.

Turbonomic supports 'parking' actions for cloud resources. These actions *stop* your cloud resources for a given period of time to help you reduce your cloud expenses, and then *start* these resources later when you need them. Use the Parking endpoint to get a list of parkable entities and execute parking actions on demand.

You can also enforce parking actions according to a schedule. In the Schedules endpoint, use /schedules/timespans/ requests to configure parking schedules and attach the schedules to parkable entities. When parking schedules have been attached to entities, use the Parking endpoint to pause and resume schedules as needed.

• **Improvement:**

Fix Version: 8.9.2

The API now allows an operator field in the instanceType and newInstanceType action filters.

The API now allows an operator field in the instanceType and newInstanceType action filters. The operator can be "EQ" (equals) or "NEQ" (not equal to); if not set, the action filter uses "EQ".

Reference: API Deprecation History

The following endpoints, requests, DTOs, and/or parameters have been deprecated in the Turbonomic API. The changes were finalized and removed from the API in the listed release.

Note:

See the instructions for alternative usage for deprecated items.

8.9.1

Endpoints	None
Requests	None
Request Parameters	None
DTOs	<ul style="list-style-type: none"> • DTO: ScenarioChangeApiDTO <p>Instruction: Use the loadChanges or configChanges parameters in the ScenarioApiDTO.</p>
DTO Parameters	<ul style="list-style-type: none"> • DTO: ScenarioApiDTO <p>Parameter Name: changes</p> <p>Instruction: Use the loadChanges or configChanges parameters.</p>

8.8.1

Endpoints	None
Requests	<ul style="list-style-type: none"> • GET /search/market/ <p>Instruction: Use GET /markets.</p> <ul style="list-style-type: none"> • GET /search/target/ <p>Instruction: Use GET /targets.</p> <ul style="list-style-type: none"> • GET /groups/<groupId>/actions/<actionId> <p>Instruction: Do not use this call (not implemented).</p> <ul style="list-style-type: none"> • GET, POST /target/<targetId>/stats

	<p>Instruction: Do not use this call (not implemented).</p> <ul style="list-style-type: none"> • GET /businessunits/<busUnitId>/stats <p>Instruction: Use GET /entities/<entityId>/stats or GET /stats/<entityId>/stats. These calls use the business unit entity ID.</p> <ul style="list-style-type: none"> • POST /businessunits/<busUnitId>/stats <p>Instruction: Use POST /stats/<entityId>/stats or POST /stats.</p>
Request Parameters	None
DTOs	None
DTO Parameters	None
ENUM Values	<ul style="list-style-type: none"> • entityType:VMSpec Instruction: Use VirtualMachineSpec. • entityTypes:VMSpec Instruction: Use VirtualMachineSpec. • entityTypeFilter:VMSpec Instruction: Use VirtualMachineSpec. • providerEntityType:VMSpec Instruction: Use VirtualMachineSpec. • relatedEntityType:VMSpec Instruction: Use VirtualMachineSpec. • providerType:VMSpec Instruction: Use VirtualMachineSpec.

8.7.1

Endpoints	None
Requests	<ul style="list-style-type: none"> • DELETE /entities/<entityId>/tags/<tagKey> Instruction: Use DELETE /entities/<entityId>/tags?key=<tagKey>. • DELETE /groups/<groupId>/tags/<tagKey> Instruction: Use DELETE /groups/<groupId>/tags?key=<tagKey>.
Request Parameters	None
DTOs	<ul style="list-style-type: none"> • LicenseApiInputDTO Instruction: Use LicenseApiDTO.
DTO Parameters	<ul style="list-style-type: none"> • DTO: HttpProxyDTO Parameter Name: portNumber Instruction: Use proxyPortNumber. • DTO: LogEntryApiDTO

	Parameter Name: <code>reasonCommodity</code> Instruction: Use the array, <code>reasonCommodities</code> .
--	--

8.6.1

Behavior Changes:

- Pagination of Returned Data from `/search` requests:

For searches, the API paginates the returned data. Starting with version 8.6.1, the default pagination limit is 100 entries, and the maximum pagination limit is 500. If you do not specify a pagination limit, then the API returns pages set to the default limit of 100. If you make no settings, and your search results include more than 100 entries, the results will be paginated. Your scripts should always check for a pagination cursor when processing search results.

You can change these settings in the `charts_v1alpha1_xl_cr.yaml` resource. Find the entries for:

- `apiPaginationDefaultLimit`
- `apiPaginationMaxLimit`

Endpoints	None
Requests	None
Request Parameters	None
DTOs	<ul style="list-style-type: none"> • <code>ReportScheduleApiDTO</code> This release will remove <code>ReportScheduleApiDTO</code>.
DTO Parameters	<ul style="list-style-type: none"> • DTO: <code>ActionApiDTO</code> Parameter Name: <code>CurrentLocation</code> Instruction: The object in this parameter will only contain the following fields: <ul style="list-style-type: none"> - <code>links</code> - <code>uuid</code> - <code>className</code> - <code>displayName</code> - <code>discoveredBy</code> - <code>environmentType</code> - <code>vendorIDs</code> • DTO: <code>ActionApiDTO</code> Parameter Name: <code>NewLocation</code> Instruction: The object in this parameter will only contain the following fields: <ul style="list-style-type: none"> - <code>links</code> - <code>uuid</code> - <code>className</code> - <code>displayName</code> - <code>discoveredBy</code> - <code>environmentType</code>

	<ul style="list-style-type: none"> - vendorIDs • DTO: ProductVersionDTO Parameter Name: buildUser Instruction: Deprecated with no replacement. This parameter was not implemented in the 8.x product family. • DTO: GroupApiDTO Parameter Name: remoteId Instruction: Deprecated with no replacement. This parameter was not implemented in the 8.x product family. • DTO: UserApiDTO Parameter Names: <ul style="list-style-type: none"> - roleUuid - roleName Instruction: Deprecated. Use the roles parameter in UserApiDTO.
--	---

Versions Older than 8.6.0

To download documentation for versions older than 8.6.0, see the [Archives](#).

Reference: API Deprecation Contract

The following endpoints, requests, DTOs, and parameters are deprecated in the Turbonomic API. The changes will be finalized and removed from the API in the listed release.

Note:

See the instructions for alternative usage for deprecated items.

8.10.1

Endpoints	None
Requests	None
Request Parameters	None
DTOs	None
DTO Parameters	<ul style="list-style-type: none"> • DTO: StatApiInputDTO Parameter Name: costPrice Instruction: Use the dedicated cost endpoints in /entities, /groups, and /markets instead. <p>Note: These are marked "In Development" currently but will reach GA soon.</p> <ul style="list-style-type: none"> - /entities/{entity_Uuid}/cost - /groups/{group_Uuid}/cost - /markets/{market_Uuid}/cost

8.11.1

Endpoints	<ul style="list-style-type: none">• GET /tags/{tagKey}/entities <p>Instruction: Use GET /tags/entities. This endpoint better supports searching for certain special characters (like slash) within tag names.</p>
Requests	None
Request Parameters	None
DTOs	None
DTO Parameters	None

Chapter 3. Installation

This guide gives you information you need to install Turbonomic in your virtual environment, install your license, and get started managing your resources.

If you have any questions, please contact IBM support. Visit our support site [here](#).

Minimum Requirements

License Requirements

To run Turbonomic on your environment, you must install the appropriate license. Licenses enable different sets of Turbonomic features, and they support a specified number of workloads in your environment.

User Interface Requirements

To display the Turbonomic user interface, you must log into the platform with a browser that can display HTML5 pages. Turbonomic currently supports the following browsers:

- Apple Safari
- Google Chrome
- Microsoft Edge
- Mozilla Firefox

Network Addressing Requirements

Turbonomic requires static IP addressing. Static IP setup is covered as a step when installing the Turbonomic VM image.

Compute and Storage Requirements

The requirements for running a Turbonomic instance depend on the size of the environment you are managing. Turbonomic keeps a real-time representation of your environment in memory. The greater the number of entities to manage, and the more extensive the relationships between them, the more resources you need for the VM that runs Turbonomic. And as the VM requirements increase, so do the requirements for the physical machine that hosts the VM.

Note:

The recommended requirements needed to deploy the SaaS client differ from what is documented here. See [OVA: Installing the vCenter Image for the SaaS Client](#).

The requirements listed here are recommendations that you should keep in mind as you plan your Turbonomic deployment. After deploying, if you find that you need to change memory capacity, CPU capacity, or both for the VM, you can shut it down, make changes, and then power it up again to use the new capacity.

Note:

The machine that hosts the Turbonomic platform must support the SSE4.2 instruction set. Support for this instruction set was introduced at different times for different chip manufacturers:

- Intel: November 2008
- AMD: October 2011

The machine you use to host Turbonomic should be newer than these dates. On a Linux system, you can run the following command to check for this support:

```
cat /proc/cpuinfo | grep sse4
```

For more information, see the glossary entry at <http://www.cpu-world.com/Glossary/S/SSE4.html>.

In most cases you can run Turbonomic on a host that meets the following minimum requirements:

Supported VM Image Technology	Storage Requirements	Memory	CPUs
VMware vCenter versions 7.0 and 8.0	1.25 TB or greater. Note:	• Default: 128 GB • For 10,000 VMs or less, 64 GB	8 vCPUs
Microsoft Hyper-V Server 2012 R2 or later	Can be thin provisioned depending on the storage requirements.		

IBM provides a VM image (an OVA or VHD file) which is preconfigured with two hard drives. A minimum of 1.25 TB is necessary to ensure that the drives have the proper amount of space for storage.

Installing on a Virtual Machine Image

You can get a download of the Turbonomic platform as a:

- VMware OVA 1.0 image
- Microsoft Hyper-V image

Note:

For minimum requirements, we recommend 128 GB of memory for the VM that hosts Turbonomic. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See [Minimum Requirements](#)).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See [Deploy the Turbonomic VM](#)).

You will install the platform in two main steps:

1. Install the Turbonomic VM image on your network.

This installs and starts up the VM that will host your instance of the Turbonomic platform.

2. Deploy the Turbonomic components on the VM.

About the Turbonomic VM Image

Turbonomic installs as a VM that runs the CentOS Linux OS. For each new version, we deliver a VM image (OVA or VHD) that you install to run the product. Typically you install this image once, and for subsequent updates to Turbonomic you will execute product updates on that installed VM. This means two things:

- Product updates patch new components of the Turbonomic application stack onto the same CentOS platform that you got when you originally installed the VM image. Product updates do not affect the underlying OS.
- Over time, you might learn of important security patches for the CentOS distribution. It is your responsibility to keep the OS up to date. You can install these patches on your Turbonomic VM whenever necessary.

Note:

We currently release the VM image with the CentOS Linux OS. We have found it to meet overall security requirements. We intend to continue with CentOS for as long as that platform remains viable and secure.

Turbonomic Updates

Because IBM continually and rapidly innovates and improves all aspects of this product, newer versions of this product are released every two weeks. For more information about installing updates, see [Updating Turbonomic to a New Version](#).

Turbonomic installs as a Kubernetes cluster on the VM. When you do update the product, the update can change:

- The application stack

This comprises a number of components and services such as:

- Application services for the user interface, analysis, and other product components.
- The historical database.
- Probes – Components that implement the different targets you can configure.

For a full list of components in the application stack, you can inspect the platform's `cr.yaml` file.

- Datastores

The installation configures persistent and ephemeral storage for the Turbonomic product to use.

- Cluster configuration and management services

This includes Kubernetes Operator and associated Kubernetes configuration files.

OVA: Installing the vCenter Image for On-prem Environments

The first step to installing Turbonomic is to deploy the VM that will host the platform. For vCenter environments, we deliver an OVA image for each bi-weekly release.

Note:

The OVA image can be installed in your on-prem environment as:

- A standalone platform that manages both on-prem and cloud targets
- A lightweight SaaS client that enables a SaaS server in the public cloud to manage on-prem targets

This topic outlines the steps to install the image as a standalone platform. For instructions on installing the image as a lightweight SaaS client, see [Installing the vCenter Image for the SaaS Client](#).

Note:

For minimum requirements, we recommend 128 GB of memory for the VM that hosts Turbonomic. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See [Minimum Requirements](#)).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See [Deploy the Turbonomic VM](#)).

To install the Turbonomic OVA:

1. Download the Turbonomic installation package.

The email you received from IBM includes links to the Turbonomic download pages. You can get the installation package from there.

The installation package includes the `turbonomic_t8c-<version>-<XXXXXXXXXXXXXXXX>.ova` file where `<version>` is the Turbonomic version number and `<XXXXXXXXXXXXXXXX>` is the timestamp.

For example: `turbonomic-t8c-8.0.5-20190916164429000.ova`

The OVA file deploys as a VM with the Turbonomic components already installed.

2. Import the OVA file into your datacenter.

Use the vCenter Server client to import the OVA into your environment.

3. Deploy the Turbonomic VM.

Configure the VM that was deployed from the OVA file.

If you want to deploy a VM with 64 GB of memory, manually modify the default value for Memory:

- a. Right-click the VM and choose **Edit Settings**.
 - b. Type **64** for Memory.
 - c. Click **OK** to save the settings
 - d. Power on the VM.
4. Open the remote console.

For the Turbonomic VM that you just deployed:

- a. Choose the **Summary** tab.
 - b. Click **Launch Remote Console**.
5. Set up the Turbonomic System Administrator account.
- a. In the remote console, log in with the following default credentials:

- Username: turbo

Do not use the account name, root.

- Password: vmturbo

Then, you will be prompted to enter a new password.

- b. Enter your new password.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

Note:

Be sure to save the changes account credentials in a safe place. For security reasons, this is the only account that can access and configure the Turbonomic VM.

- c. Enter your new password again to verify it.

6. Update the root password.

The platform uses the root account for certain processes, such as rolling up log messages in `/var/log/messages`. To ensure the account credentials are current, you must change the password:

- a. Open a SuperUser session.

- In the remote console, enter `su -`
- At the password prompt, enter the default password: `vmturbo`

- b. Reset a new password.

After you log in as root with the default password, the system prompts you for a New password. This new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

Note:

Be sure to save the root account credentials in a safe place.

- c. Exit the SuperUser session.

Enter `exit`.

7. Set up the static IP address.

Operation and management of the Turbonomic platform both require a static IP address for the VM. The installed VM includes the `ipsetup` script to perform this task. To run the script, execute the following:

```
sudo /opt/local/bin/ipsetup
```

When the script runs it requests the following inputs.

Note:

You must provide values for these required fields. Otherwise the installation can fail or your VM can be unreachable:

- **Required:** Do you want to use DHCP or set a static IP...
Choose static
- **Required:** Please enter the IP Address for this machine
- **Required:** Please enter the network mask for this machine
- **Required:** Please enter the Gateway address for this machine
- **Required:** Enter DNS Server(s) IP Address for this machine

You should make a note of the IP address that you provide.

As part of running `ipsetup`, you can configure a proxy for internet access. To verify that your proxy has been properly set, first complete the full installation procedure. After you run the installation you can view the proxy configuration in `/etc/systemd/system/containerd.service.d/http-proxy.conf`. Find the `[Service]` section. It should contain your proxy and `no_proxy` settings. For example, it should be similar to:

```
Environment="HTTP_PROXY=http://10.10.12.34:3123" "NO_PROXY=<YourStaicIP>, node1, 127.0.0.1, 127.0.0.0"
```

8. Perform other necessary configuration steps, and then install the Turbonomic components.

To perform the required and important configuration steps for the Turbonomic instance, see [General Configuration Tasks](#).

To install the Turbonomic components, see [Deploying the Turbonomic Components](#).

VHD: Installing the Microsoft Hyper-V Image

The first step to installing Turbonomic is to deploy the VM that will host the platform.

For Hyper-V environments, we deliver a Hyper-V image for each quarterly release. If you want to run Turbonomic on a Hyper-V VM, you can install the Quarterly Release, and then update to a later point release if necessary.

Note:

For minimum requirements, we recommend 128 GB of memory for the VM that hosts Turbonomic. However, if you plan to manage a smaller environment (10,000 VMs or less), you can install on a VM that provides 64 GB of memory. (See [Minimum Requirements](#)).

If you plan to install a VM with 64 GB of memory, then you must modify the default for VM memory. (See [Deploy the Turbonomic VM](#)).

To install Turbonomic:

1. Download the Turbonomic installation package.

The email you received from IBM includes links to the Turbonomic download pages. You can get the installation package from there.

2. Expand the .zip file and copy the contents, which includes the Virtual Machine image, to your Hyper-V server (either to your cluster shared volume or to a local hard drive).
3. Use the Import Virtual Machine Wizard in the Hyper-V Manager to import the Virtual Machine into your environment.
4. Make sure your virtual network adapter is connected to the correct virtual network.
5. Ensure the Turbonomic instance will have sufficient memory.

IBM recommends that you use static memory for your Turbonomic instance. However, you can specify static or dynamic memory for the instance. By default, the installation sets static memory to 128 GB.

6. Start the Turbonomic appliance and record its IP address.
7. Set up the Turbonomic System Administrator account.
 - a. Log into the VM's Hyper-V console with the following default credentials:

- Username: turbo
- Password: vmturbo

Then, you will be prompted to enter a new password.

- b. Enter your new password.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

Note:

Be sure to save the changed account credentials in a safe place. For security reasons, this is the only account that can access and configure the Turbonomic VM.

- c. Enter your new password again to verify it.

8. Update the root password.

The platform uses the root account for certain processes, such as rolling up log messages in /var/log/messages. To ensure the account credentials are current, you must change the password:

- a. Open a SuperUser session.
 - In the remote console, enter `su -`
 - At the password prompt, enter the default password: `vmturbo`
- b. Reset a new password.

After you log in as root with the default password, the system prompts you for a New password. This new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

Note:

Be sure to save the root account credentials in a safe place.

- c. Exit the SuperUser session.

Enter `exit`.

9. Set up the static IP address.

Operation and management of the Turbonomic platform both require a static IP address for the VM. The installed VM includes the `ipsetup` script to perform this task. To run the script, execute the following:

```
sudo /opt/local/bin/ipsetup
```

When the script runs it requests the following inputs.

Note:

You must provide values for these required fields. Otherwise the installation can fail or your VM can be unreachable:

- **Required:** Do you want to use DHCP or set a static IP...
Choose static
- **Required:** Please enter the IP Address for this machine
- **Required:** Please enter the network mask for this machine
- **Required:** Please enter the Gateway address for this machine
- **Required:** Enter DNS Server(s) IP Address for this machine

You should make a note of the IP address that you provide.

As part of running `ipsetup`, you can configure a proxy for internet access. To verify that your proxy has been properly set, first complete the full installation procedure. After you run the installation you can view the proxy configuration in `/etc/systemd/system/containerd.service.d/http-proxy.conf`. Find the `[Service]` section. It should contain your proxy and `no_proxy` settings. For example, it should be similar to:

```
Environment="HTTP_PROXY=http://10.10.12.34:3123" "NO_PROXY=<YourStaticIP>, node1, 127.0.0.1, 127.0.0.0"
```

10. Enable the NIC for the installed VM.

The Turbonomic instance configuration includes one NIC, but it is not enabled or connected to a network. Display the NIC in the Hyper-V Manager and enable it.

11. Perform other necessary configuration steps, and then install the Turbonomic components.

To perform the required and important configuration steps for the Turbonomic instance, see [General Configuration Tasks](#).

To install the Turbonomic components, see [Deploying the Turbonomic Components](#).

Deploying the Turbonomic Components

After you install the Turbonomic VM that will host the platform, you can install the platform components.

First, gather the information you will need to run the installation:

- Network Time Source for Time Synchronization (optional)

You can perform this step during installation, or at a later date. If you want to synchronize the VM's clock now, you will be prompted for the Network Time Source. For more information about synchronizing the VM's clock, see [Synchronizing Time](#).

- Your updated root password

The installation script requires that you have updated the `root` password for the VM. If you followed the instructions in [OVA: Installing the vCenter Image](#) or in [VHD: Installing the vCenter Image](#), then you should have already performed this step.

When you are ready with the necessary information, you can run the installation script.

The IBM License Service, which collects license usage information that is used to create usage snapshots and reports, is installed automatically as part of the `t8cinstall.sh` script. For more information, see [Tracking License Usage with IBM License Service](#).

The following steps describe the default installation process. If you want to customize your installation, then you should consider taking the steps in [Stepwise Platform Deployment](#). For example, to change the Kubernetes host name for the deployment, you must perform a stepwise installation.

Note:

Starting with Turbonomic version 8.7.5, IBM Container Registry is used for all Turbonomic images for online upgrades and new installs. Ensure you have access to <https://icr.io> before continuing.

All OVA installs use Kubernetes v1.24.6. Docker commands no longer work. You can use `crictl` and `ctr` commands instead. For more information, see the following resources:

- [Container runtime changes in Kubernetes 1.24 and beyond](#) on the Kubernetes documentation site
- [Mapping from dockercli to crictl](#) on the Kubernetes documentation site
- [crictl command reference and information](#) on GitHub
- [ctr command reference and information](#) on GitHub

1. Start up the installation script.

- Start a secure session (SSH) on your Turbonomic VM as the `turbo` user.
- Execute the installation script:

```
sudo /opt/local/bin/t8cInstall.sh
```

2. Verify that you have configured a static IP address for the Turbonomic VM.

After the components start up, you will type this static IP address into a web browser to access the login page for the Turbonomic user interface.

As a first step, the script prompts you with:

```
Have you run the ipsetup script to setup networking yet? [y/n] n
```

If you have not configured a static IP for the platform VM, enter `n` to exit the installation script now, and configure a static IP.

If you have already configured a static IP for the platform VM, enter `y` to continue the installation. The script output displays the IP address that it recognizes for the VM, for example:

```
-----  
Old IP Address: 10.0.2.15  
New IP Address: 10.10.123.123  
-----
```

Because of dependencies between Turbonomic and the Kubernetes installation, it is not recommended to change the IP address after the Turbonomic installation. For a production installation of Turbonomic, the VM must run with a static IP. For a testing or evaluation installation, you can use DHCP. However, if you plan to later use such an installation in a production environment, be sure to configure a static IP.

If you followed the instructions in [OVA: Installing the vCenter Image](#) or in [VHD: Installing the vCenter Image](#), then you already ran the `ipsetup` script to configure a static IP.

3. Wait while the script performs the installation.

As the installation process continues, the script:

- Configures the platform environment with the necessary certificates
- Configures the Kubernetes cluster on the VM

This can take a few tries before it succeeds. For each try that does not succeed, you will see messages similar to:

```
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

When the connection succeeds, the script advances to the next steps.

- Establishes local storage for the platform
- Creates the kubernetes namespace for the platform as `turbonomic`

- Configures authorization to access the required Kubernetes secrets
- Initializes the MariaDB database server to manage historical data for the platform

The script creates two accounts on the MariaDB that have full privileges:

– root@localhost

This account does not use a password. To connect via this account the user must be system root.

– mysql@localhost

This account does not use a password. To connect via this account the user must be system mysql.

Note:

For security reasons, IBM recommends that you configure passwords for these accounts. You can connect with these accounts via sudo. For example, sudo mysql. After you connect, you can then set passwords to these accounts. For more information, see the MariaDB Knowledgebase at <https://mariadb.com/kb>.

- Installs the Timescale database for Embedded Reports and the Data Exporter
- Deploys and starts up the platform components

As the deployment begins, the script prints out the following:

```
#####
Start the deployment rollout
#####
```

After it deploys the components, it waits for the components to start up:

```
The installation process is complete, waiting for all the components to start up.
** The script will wait for as long as 30 minutes. **
```

If the components all start up within 30 minutes, then the installation is complete and successful.

If the components do not all start up within 30 minutes, the script displays the following and then exits:

```
=====
One or more of your deployments has not started up yet.
** Please give your environment another 30 minutes to stabilize. **
To check the status of your components, execute the following command:
kubect1 get pods
If some components are still not ready, contact your support representative
Deployments not ready:
```

The script then displays the formatted result of the kubect1 get pods command. This shows you the current status of the pods in the Turbonomic platform.

If the script exits before the components all start up, we recommend that you give the platform another 30 minutes. To periodically check the component status, run the kubect1 get pods command. If the components do not all start up after another 30 minutes, contact your support representative.

If the installation is successful and the components are all started up, the script displays a message similar to the following, where it gives the VM's static IP address:

```
#####
Deployment Completed, please login through the UI
https://10.10.123.123
#####
```

You can move on to the next steps.

4. Save a copy of the platform's Master Key secret.

The installation procedure creates a Master Key secret in the Kubernetes cluster. Turbonomic uses this secret to provide access for the platform components. You should save the key data to a safe location. If for some reason the key data gets corrupted or is otherwise unusable, Turbonomic will fail to operate. If this happens, you can contact your support representative and use this saved data to recover your platform.

To save the data:

- a. List the platform secrets.

Execute the command:

```
kubectl get secrets
```

The results should include the Master Key secret, similar to the following:

```
...
master-key-secret          Opaque          1          57d
...
```

- b. Display the Master Key data.

Once you find the Master Key name, you can then display the key data:

```
kubectl get secret master-key-secret -o yaml
```

The command result should be similar to the following:

```
apiVersion: v1
data:
  primary_key_256.out: AfnJWutxNHAduaIOdAii3DRA2fMa6lzX4rWetZxxZvc=
kind: Secret
metadata:
  creationTimestamp: "2021-06-30T02:59:19Z"
  managedFields:
  - apiVersion: v1
    fieldsType: FieldsV1
    fieldsV1:
      f:data:
        .: {}
        f:primary_key_256.out: {}
      f:type: {}
    manager: kubectl-create
    operation: Update
    time: "2021-06-30T02:59:19Z"
  name: master-key-secret
  namespace: turbonomic
  resourceVersion: "1072"
  uid: a314b2ba-2061-4b41-b844-56caf2c3728d
type: Opaque
```

The important key data to save is the `primary_key . . . data`. In the above example, you should save the line:

```
primary_key_256.out: AfnJWutxNHAduaIOdAii3DRA2fMa6lzX4rWetZxxZvc=
```

- c. Save the data to a safe place.

Write this data to a file and save it in a safe backup location. If you ever need to recover the Master Key, your support representative will use this data to perform the recovery.

5. Log in to the Turbonomic user interface and set the administrator user account password.

Turbonomic includes a default user account named `administrator` which has an `ADMINISTRATOR` role. As you log in for the first time, you must set your own password for that account. You can create or delete other accounts with the `ADMINISTRATOR` role, but your installation of Turbonomic must always have at least one account with that role.

In the login page, enter the information as required, and make a note of it.

- Use the default credential for **USERNAME**: `administrator`.

- Type a password for **PASSWORD**.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

- Type the password again to verify it for **REPEAT PASSWORD**.
- Click **Create Account**.

This is the account you will use to access the Turbonomic user interface with administrator permissions. Be sure to save the user interface administrator account credentials in a safe place.

Note:

The initial login is always for the default user account named administrator which has an ADMINISTRATOR role.

6. After you have logged in as administrator, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see [Managing User Accounts](#).

Note:

For security reasons, you can create a different account with an ADMINISTRATOR role to serve as the main administrator of your Turbonomic installation, and then delete the default administrator account. But remember, you must always have at least one user account with administrator privileges.

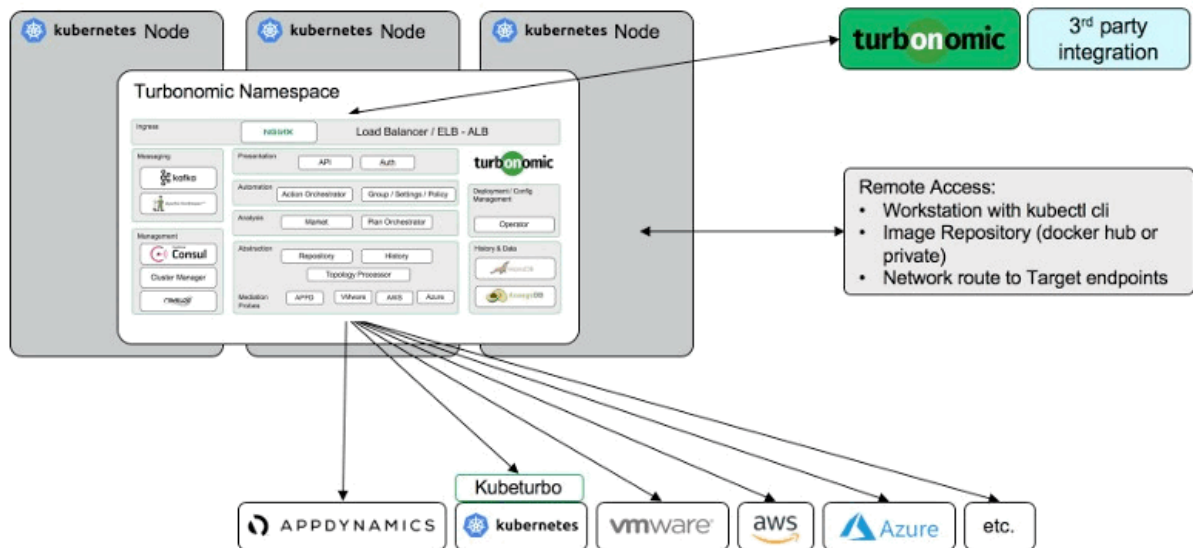
Installing on Kubernetes Clusters

Kubernetes, also known as “k8s” or “kube,” is a container orchestration platform for scheduling and automating the deployment, management, and scaling of containerized applications.

Note:

Installation on Kubernetes clusters is only supported on the Red Hat OpenShift platform.

Kubernetes knowledge is required for this installation method.



Turbonomic is a containerized microservices platform that installs onto a Red Hat OpenShift cluster in a single namespace deployment. The platform uses an Operator pattern to deploy, manage, and run the IBM components. The Turbonomic Platform Operator (t8c-operator) uses a Custom Resource Definition (kind: XI), and uses a Custom Resource to configure and adjust your setup.

As part of your deployment, you can take advantage of the backup and HA capabilities of an external database service running outside of the node cluster.

Installing on Red Hat OpenShift Clusters

Supported versions

- Red Hat OpenShift 4.8.x or higher

Installation methods

To install the Turbonomic Server on Red Hat OpenShift, we recommend installing a Turbonomic Platform Operator from OperatorHub using the Red Hat OpenShift Container Platform console as this is the certified and tested method recommended by Red Hat.

Note:

The user deploying the Turbonomic Platform should have sufficient privileges to create a namespace and deploy from the OperatorHub. If you are airgapped or running a disconnected cluster, ensure you can use the OperatorHub by following Red Hat documentation applicable for your version of the Red Hat OpenShift Container Platform. See [Using Operator Lifecycle Manager on restricted networks](#) in the Red Hat OpenShift documentation for details.

Planning the deployment

Review the [prerequisites](#) and [resource recommendations](#) before proceeding with installing the Operator.

Note:

The Turbonomic Platform Operator will install the latest version in a single namespace. Before installation, you should decide which approval strategy you want: Automatic or Manual. See [Understanding OperatorHub](#) and [Installing Operators in your namespace](#) in the Red Hat OpenShift documentation for details.

You should also review the following options before deployment, which include services configured outside of Turbonomic. If you want to use one of these setup options, review the links for more details.

- Private container registry: See [Working with a Private Repo & Image Pull Secrets](#)
- [Configuration Options](#) contains other common setup options, most of which can also be configured post deployment

Remote database server configurations

Turbonomic supports a remote database for historical data. This configuration allows for high availability, backups, and handling larger IO demands for larger environments or from remote SQL Server queries (for reporting and so on).

Note:

Before deploying Turbonomic, you must configure a remote database server for historical data. If you are not going to use an existing database server, you must first create a new database server and then configure it as noted in this section.

1. If creating a new database server to use with Turbonomic, create a remote database using the recommended settings for a MariaDB or MySQL instance as noted below.
 - Preference: MariaDB 10.5.20; MariaDB 10.2 for Azure DB Services is tolerated. MySQL 5.7 is also supported.
 - Cluster configuration: master/replica where Turbonomic communicates to a single replica
 - MariaDB 10.5 requires 10.5.20 minimally

Important:

Because of a known issue, *you must never use* MariaDB versions 10.5.14, 10.5.15, 10.6.7, 10.7.3, or 10.8.2.

- Select network and security policies that allow communication between the Kubernetes cluster and the Database server; the default DB port is 3306
 - Minimum memory: 8 GB memory/2 vCPU
 - Storage: 100 GB (set to auto-increase) or provide 1024 GB (or 1TB)
 - Encryption (SSL) is supported
2. Configure the Database service's server configuration (in Amazon Web Services (AWS), create a new RDS Parameter Group). Database configuration should start from the default configuration, and then include the specific parameters listed below.

Parameter	Value
log_bin_trust_function_creators	1
event_scheduler	on
max_allowed_packet	1GB
sql_mode	ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER, NO_ENGINE_SUBSTITUTION
explicit_defaults_for_timestamp	off
wait_timeout	28800
character_set_client, character_set_connection, character_set_database, character_set_results, character_set_server	utf8

3. If you are working with your own DB Server / DB Service, create a Turbonomic admin user that will be used by the deployment to create application users and grant privileges. You can define your own password, but it should be reflected in the Custom Resource YAML. Using a MySQL client or CLI, run the following commands, substituting your credentials, against your database server:

```
CREATE USER 'turboadmin'@'%' IDENTIFIED BY 'vmturbo';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES, INDEX,
ALTER, SHOW DATABASES, CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION SLAVE,
REPLICATION CLIENT, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER ROUTINE, CREATE USER,
EVENT, TRIGGER ON *.* TO 'turboadmin'@'%' WITH GRANT OPTION;
```

4. If you are using AWS RDS or an Azure DB service, see [AWS RDS DB Service –Additional Configuration](#) or [Azure DB Service –Additional Configuration](#). You will need to create the application database users, databases (empty schema), and grant privileges.

The database server should now be ready to be included in your Turbonomic installation.

Note:

For information on using Kubernetes secrets for DB app username and password credentials, see [Kubernetes Secrets for DB Credentials](#).

Prerequisites

Install the Turbonomic platform on an existing Red Hat OpenShift cluster. The following is a checklist of requirements and additional configurations that will require modification of the default Custom Resource YAML (cr.yaml). To support this cluster, your environment must support the following prerequisites:

	Requirements	Refer to
Supported Version	Red Hat OpenShift 4.8.x or higher, Linux x86 architecture only	
Access	<ul style="list-style-type: none"> • Admin privileges on the OpenShift cluster • OC CLI • Optional MariaDB or MySQL client for remote database setup 	
Resources	The amount of resources required to run Turbonomic is related to the complexity and type of entities it will manage (virtual machines, containers/pods, cloud accounts, and so on) in your environment. See Resource Recommendations .	
Server Version	The server version to deploy must be specified in the Custom Resource YAML during installation.	Installation step “5.b” on page 42
Remote Database (required for production environment)	<p>A remote database is a database server with backup and HA capabilities. Turbonomic currently supports MariaDB version 10.5.20 and MySQL 5.7. This support includes comprehensive testing and quality control for Turbonomic usage of the database.</p> <p>Important:</p> <p>Because of a known issue, <i>you must never use</i> MariaDB versions 10.5.14, 10.5.15, 10.6.7, 10.7.3, or 10.8.2.</p> <p>DB Server should be configured with a minimum of 8 GB RAM, 4 Cores/CPU, and 1 TB disk/file system space for tables. See Using a Database Server or Service.</p> <p>Note:</p> <p>Using a containerized database is not supported as it can lead to irrecoverable data corruption. Additionally, a containerized database's performance is directly affected by network performance and cannot be guaranteed. A containerized database should be used for demo, Proof of Concept (POC), or testing purposes only.</p>	Remote database server configurations and installation step “5.c” on page 42
Security Context	<p>Components need read/write access to Persistent Volumes (PVs) and may use a range of fsGroup ids. Where required, security context for which UIDs are to be used for a pod to write to its PV is configured using the project's scc uid, obtained from the project's properties and supplied to the Turbonomic deployment through the Custom Resource YAML.</p> <p>Note:</p> <p>Security Context is a requirement for Red Hat OpenShift.</p>	Installation steps “3” on page 41 and “5.d” on page 43

	Requirements	Refer to
Storage	<p>The Turbonomic platform leverages Persistent Volumes (PVs) to store logs, certificates, and tokens, and uses Persistent Volume Claims (PVCs) to create PVs from the cluster's default storage class, or a Security Context that is specified in the Turbonomic Custom Resource.</p> <p>Best Practices: Volume backup strategy is important. The PVs associated with the following PVCs are ones that should also be backed up to ensure recovery in case of a loss of the server: api, api-certs, auth, topology-processor, and consul-data. If you are running a containerized database for Turbonomic, also include: db-data and data-timescaledb-0.</p> <p>The following Storage Class properties are required:</p> <ul style="list-style-type: none"> • To support pods with multiple PVs, especially where your nodes are spread across multiple AZs or regions, you must set binding to be <code>volumeBindingMode = WaitForFirstConsumer</code>. • Ensure recovery by setting the <code>reclaimPolicy</code> option of <code>Retain</code>. • Ideally, also use a storage provisioner that allows for volume expansion (<code>allowVolumeExpansion</code>). <p>See Storage Class Requirements for PVC details and total storage required.</p>	
Network	<p>The operator creates an NGINX LoadBalancer service by default, which uses an IP address or a load balancing service (external or internal), depending on the annotation. Public IP/ELB are default cloud load balancers.</p> <p>Since NGINX should be deployed as a proxy service, you must edit the Custom Resource YAML to include NGINX attributes. Additionally, you can choose to have Turbonomic create a route in front of nginx or you can manually create the route.</p> <p>See NGINX Service Configuration Options; for Red Hat OpenShift, see Platform Provided Ingress.</p>	Installation step “5.e” on page 43
Container Images	<p>We support both public and private container repositories. By default, the Turbonomic installation uses the public IBM Container Registry.</p> <p>See Using Operator Lifecycle Manager on restricted networks in the Red Hat OpenShift documentation if you are airgapped or running a disconnected cluster and still want to use OperatorHub.</p> <p>All methods may configure the deployment to pull from a private repository and use pull secrets. See Working with a Private Repo and Image Pull Secrets.</p>	Installation steps “5.b” on page 42 and “5.f” on page 44

	Requirements	Refer to
Desired Targets	<p>Targets are not enabled by default. For a full description of supported targets, see Target Configuration.</p> <p>Note:</p> <p>Targets can be enabled or disabled through Probe settings in the Custom Resource YAML at any time during or after the installation is complete.</p>	Installation step “7” on page 44

Resource Recommendations

The Turbonomic platform is a micro service-based application. It takes advantages of containers to scale the platform to handle anywhere from small to huge IT estates. The amount of resources you need to run Turbonomic is related to the complexity and type of entities it will manage. The following table provides a guideline to cluster resources you might need to support Turbonomic for different numbers of entities you expect to manage in a single instance:

Entity Scale	Minimum Node Size for Node Group	Number of Nodes	Approx Total Mem Consumed (GB)	Node Storage (GB)
Standard (under 80K)	2 CPUs, 32 GB Memory	3+	32-64	10
Large (80K plus)	4 CPUs, 32 GB Memory	4+	32-128	10
X-Large (200K plus)	contact Turbonomic			

Note:

- Persistent Volumes (PVs) are created based on dynamic storage provisioning. See PV requirements [here](#).
- The total amount of memory and CPUs required also depends on the number of mediation components enabled, and the total number of configured targets. Each target type spins up its own pod.
- Turbonomic supports namespaces with resource quotas. Contact Turbonomic Support to help configure a quota.

Installing using the Red Hat OpenShift OperatorHub

Turbonomic installations use an Operator to deploy and manage the Turbonomic instance. This topic provides instructions for deploying the Turbonomic Platform in Red Hat OpenShift 4.8.x or higher using the Red Hat OpenShift OperatorHub.

Note:

Starting with Turbonomic version 8.7.5, IBM Container Registry is used for all Turbonomic images for online upgrades and new installs. Ensure you have access to <https://icr.io> before continuing.

This installation procedure includes creating an instance of the platform Operator, which defaults to `x1-release`, and configuring the Custom Resource YAML for your desired configuration.

Note:

It is strongly recommended that you deploy Kubeturbo after deploying the Turbonomic server so that we can monitor Red Hat OpenShift Container Platform clusters to optimize resource requirements based on your actual usage. Deploy the Kubeturbo operator into the monitored cluster in a separate project

(namespace) and configure the probe to communicate with the Turbonomic Platform. See the article for deploying Kubeturbo via [Red Hat OpenShift OperatorHub Details](#) for more information.

1. Create and configure a remote database for historical data. This must be done prior to setting up the Turbonomic Platform. See [Remote database configurations](#).

Note:

This skips the option to use the containerized database entirely. A containerized database should be used for demo, Proof of Concept (POC), or testing purposes only.

2. Create a new project in the Red Hat OpenShift cluster.

This project is used throughout the installation process. The Operator must be installed in this project (namespace).

- a. Navigate to **Home > Projects**.
 - b. On the Projects page, click **Create Project**.
 - c. Enter the name of the project (for example, `turbonomic`) and click **Create**.
3. Determine the Security Context Constraints (SCC) UID range. This is used later in the install to allow pods to read and write to their PVs.
 - a. Open the newly-created project from the Project page and click on the **YAML** tab.
 - b. Find and make note of the Security Context Constraints (SCC) UID range.

You will see a line similar to the one below. In this example, `1000680000` is the SCC UID value.

```
openshift.io/sa.scc.uid-range:1000680000/10000
```

4. Deploy Turbonomic into the new project using the OperatorHub.

- a. Navigate to **Operators > OperatorHub**.
- b. In the **Filter by keyword** box, enter `turbonomic` to find and select the Turbonomic Platform Operator.

You will see multiple Turbonomic Platform Operator options. To deploy Turbonomic with a certified operator, ensure you select the "Certified" edition (do not select "Community" or "Marketplace" editions).

- c. On the Turbonomic Platform Operator page, read the information about the Operator and click **Install**.
- d. On the Install Operator page, specify the Update channel, the project (Namespace) in which you want to deploy the Operator, and the desired Update approval strategy.

Note:

Ensure the `Installed` namespace is the same as the project you created earlier in the process.

We recommend that you select **Stable** as the Update channel as it is the Generally Available tested product. The beta channel is a pre-release candidate and should not be used unless instructed.

We recommend that you select **Manual** as the Update approval strategy so that you are alerted when a newer version of the Operator is available.

- e. Click **Install**.
- f. Once the Operator installs and you see "Installed operator - ready for use," click **View Operator** or select **Operators > Installed Operators** to view the Turbonomic Platform Operator.

Note:

If you selected **Manual** as the approval strategy, the upgrade status of the subscription remains "Upgrading" until you review and approve the install plan. After approving on the **Install Plan** page, the subscription upgrade status moves to "Up to date."

5. Create an instance of the Platform Operator and configure the Custom Resource YAML for your desired configuration.

For the Turbonomic Custom Resource, indentation defines where parameters are applied (globally or to specific components), so ensure you line up the text appropriately. Follow the tips in [Working with YAML Files](#).

Note:

For examples of common configurations, see our [sample custom resource YAMLs](#).

- a. On the **Details** tab for the Turbonomic Platform Operator, under **Provided APIs**, click **Create Instance**.

On the Create XI page, select the **YAML** view to display the Custom Resource YAML file editor.

Note:

The YAML file can be accessed at any time during or after configuration by navigating to **Installed Operators > Turbonomic Platform Operator > Turbonomic Platform Operator tab > xl-release**.

- b. Set the server version and repository.

Edit the Custom Resource YAML to change the `repository` parameter from the Red Hat Container Catalog to the IBM Container Registry (`icr.io/cpopen`).

Edit the Custom Resource YAML to include the version to deploy in the `tag: {version to deploy}` parameter as shown.

```
spec:
  global:
    customImageNames: false
    externalDbIP: 10.0.2.15
    repository: icr.io/cpopen/turbonomic
    tag: {version to deploy}
```

Note:

The `customImageNames` parameter must be set to `false` when working with Red Hat Operators, and the `repository` parameter must specify `icr.io/cpopen/turbonomic` as the repository location.

- c. Specify the external database properties.

Edit the Custom Resource YAML to add the properties for the external database you created prior to deployment, or remove the parameters to use a containerized database. Do one of the following:

- To use the external database IP address: Update the default `externalDbIP` value the IP address for your external DB server. If you use a fully qualified domain name for your DB Server, remove this parameter and use `externalDBName` instead.
- To use the external database name: Replace the `externalDbIP` parameter with the `externalDBName` parameter and include the database server's fully qualified domain name (for example, `myDBServer.myDomain.com`).
- To use a containerized database: Remove the `externalDbIP` or `externalDBName` parameter. If neither are present, a containerized database is created.

Note:

Using a containerized database is not supported as it can lead to irrecoverable data corruption. Additionally, a containerized database's performance is directly affected by network performance and cannot be guaranteed. A containerized database should be used for demo, Proof of Concept (POC), or testing purposes only.

The `global` section shown here is under the parent `spec` section of the YAML file. This example includes optional parameters and can be used as a guide for MySQL


```
spec:
  global:
    externalDbIP: {Edit this value with the IP address for your DB Server. If you
use a fully qualified domain name for your DB Server, remove this parameter and use
externalDBName instead. If you choose to use a containerized DB, this parameter should
be removed}
    externalDBName: {If not using externalDbIP to specify your DB Server's IP address,
use this parameter to specify your DB Server's fully qualified domain name, for example:
myDbServer.myDomain.com}
  properties:
    global:
      enableSecureDBConnection: true
      sqlDialect: MYSQL
      dbPort: 6033
      dbRootPassword: vmturbo
      dbRootUsername: turboadmin
```

Note:

Remove or comment out the `sqlDialect` parameter when using MariaDB, since MariaDB is the default database server type. Additional properties are required for AWS RDS and Azure DB Services. See documentation for configuring database app users, and provide that information here under `properties: global` in the parent `spec` section of the YAML file.

- d. Set the Security Context Constraints to allow pods to read and write to their PVs. If this value is not set properly, the pods will not start up successfully.

Edit the Custom Resource YAML to include the UID value you noted in step “3” on [page 41](#) in the `securityContext: fsGroup: parameter` as shown in the following example:

```
spec:
  global:
    customImageNames: false
    externalDbIP: 10.0.2.15
    repository: icr.io/cpopen/turbonomic
    securityContext:
      fsGroup: 1000680000
    tag: 8.9.3
```

- e. Set up Turbonomic to automatically create a Red Hat OpenShift route to the `nginx ClusterIP` type Service. The Turbonomic Operator can create a route for you that functions as the Turbonomic user interface and the endpoint for all remote probes, such as `Kubeturbo` configured on another cluster.

Edit the Custom Resource YAML to include the `nginx` information noted below. Change the default `nginx` from being a primary ingress to function as a proxy, then enable creating a route. The name of the default route that is created is `nginx`.

```
spec:
  global:
    customImageNames: false
    externalDbIP: 10.0.2.15
    repository: icr.io/cpopen/turbonomic
    securityContext:
      fsGroup: 1000680000
    tag: 8.9.3
  nginx:
    nginxIsPrimaryIngress: false
    httpsRedirect: false
  nginxingress:
    enabled: true
  openshiftingress:
    enabled: true
```

(Optional) If you want to manually create your own route to Turbonomic/nginx use only the parameters below:

```
nginx:
  nginxIsPrimaryIngress: false
  httpsRedirect: false
```

- f. (Optional) If you are working with a private registry, and not the IBM Container Registry, edit the Custom Resource YAML to define registry and repository locations where the default is IBM Container Registry, and make the necessary changes to the operator deployment YAML.

Once the images are in your private registry, edit the repository parameter in the Custom Resource YAML to point to `icr.io` as the registry location.

```
spec:
  global:
    customImageNames: false
    externalDbIP: 10.0.2.15
    repository: icr.io/cpopen/turbonomic
    securityContext:
      fsGroup: 1000680000
    tag: 8.9.3
```

If you require credentials to access your registry, you can specify either the image pull secret or the user name and password credentials in the Custom Resource YAML as shown in the examples below:

```
spec:
  global:
    externalDbIP: 10.0.2.15
    registry: myregistry
    repository: my.repo/myregistry/turbonomic
    imagePullSecret: turbocred
    securityContext:
      fsGroup: 1000680000
    tag: 8.9.3
```

```
spec:
  global:
    externalDbIP: 10.0.2.15
    registry: myregistry
    repository: my.repo/myregistry/turbonomic
    imageUsername: turbouser
    imagePassword: turbopassword
    securityContext:
      fsGroup: 1000680000
    tag: 8.9.3
```

Note:

- The `registry` parameter is required in the Custom Resource YAML to use the `imageUsername/imagePassword` or `imagePullSecret` parameters.
 - This method creates a Kubernetes secret of type `docker-registry` with the default name of "turbocred," and then applies the `imagePullSecret` parameter to the Turbonomic component deployments.
 - If using an `imagePullSecret`, a secret must be created and used instead of the `imageUsername` and `imagePassword`. These parameters should only be used when choosing not to utilize an `imagePullSecret`. See [Working with a Private Repo and Image Pull Secrets](#).
6. Click **Create** to create an instance of the Turbonomic Platform server.
7. When you first install Turbonomic using OperatorHub, target probes are disabled by default. Enable desired target probes, such as IaaS types whether on-prem or cloud.

A probe is a platform component that connects to a target. It discovers the target's entities and loads them into the Turbonomic supply chain, where actions can be executed on entities within the target environment. Turbonomic ships with many probe components that you can use to connect Turbonomic with your environment.

- a. Navigate to **Installed Operators > Turbonomic Platform Operator > Turbonomic Platform Operator** tab > **xl-release**.
- b. Edit the Custom Resource YAML to add and enable applicable targets for your instance. For a full description of supported targets, see [Target Configuration](#).

You should not enable Kubeturbo in this step; it should be deployed into the monitored cluster in a separate project (namespace) and configured to communicate with the Turbonomic Platform. See the article for deploying Kubeturbo via [Red Hat OpenShift OperatorHub Details](#) for more information.

All target probes need to be configured in the YAML file using the formatting shown in the following example:

```
spec:
  global:
    tag: 8.9.3
  ...
  aws:
    enabled: true
  azure:
    enabled: true
  dynatrace:
    enabled: true
  ...
```

c. Save and apply your changes to the platform.

8. Navigate to **Workloads > Pods** to confirm all pods are running and ready, for example 1/1, 2/2, and so on. This may take a few minutes.

If the pods do not run, refer to the information below for possible solutions.

- Ensure the Security Context Constraints UID is set correctly as noted above then restart the pods.
- If you do not have access to the repository that is specified for the container images, check your credentials.
- Ensure the PVs are created and bound to the PVCs.

9. In the Turbonomic project, click **Networking > Routes**, select the **nginx** route (or the route you manually created), and click on the **Location** field to open the URL to use for your Turbonomic server (for example, `nginx-yourNamespaceName.apps.yourOCPDomain`). This opens a new tab for the Turbonomic user interface.

Note:

This route will also be the one provided to users to access the Turbonomic user interface.

10. Log in to the Turbonomic user interface and set the administrator user account password.

Turbonomic includes a default user account named `administrator` which has an `ADMINISTRATOR` role. As you log in for the first time, you must set your own password for that account. You can create or delete other accounts with the `ADMINISTRATOR` role, but your installation of Turbonomic must always have at least one account with that role.

In the login page, enter the information as required, and make a note of it.

- Use the default credential for **USERNAME**: `administrator`.
- Type a password for **PASSWORD**.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

- Type the password again to verify it for **REPEAT PASSWORD**.
- Click **Create Account**.

This is the account you will use to access the Turbonomic user interface with `administrator` permissions. Be sure to save the user interface administrator account credentials in a safe place.

Note:

The initial login is always for the default user account named `administrator` which has an `ADMINISTRATOR` role.

11. Follow the steps in the setup wizard to import the license key and add target types.

- You can upload the license by dragging and dropping the license key file into the Enter License fly-out or browse to the license key file. Be sure to upload only .xml, .lic, or .jwt files.
 - After you successfully import the license key, add your targets. The targets available to add are based on the probes that you enabled in the YAML file. For example, if you enabled the AWS probe, you will see the AWS target type.
12. After you have logged in as administrator, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see [Managing User Accounts](#).

Note:

For security reasons, you can create a different account with an ADMINISTRATOR role to serve as the main administrator of your Turbonomic installation, and then delete the default administrator account. But remember, you must always have at least one user account with administrator privileges.

General Configuration Tasks

After you install the Turbonomic instance, you should perform the following configuration tasks:

(Required) Synchronizing Time

It is important that you synchronize the clock on the Turbonomic instance with the other devices on the same network. By default, the Turbonomic server is configured to synchronize with any one of the following time servers:

- 0.centos.pool.ntp.org
- 1.centos.pool.ntp.org
- 2.centos.pool.ntp.org
- 3.centos.pool.ntp.org

To synchronize with these servers, your installation of Turbonomic must have access to the internet. If your environment restricts internet access, then you have to configure synchronization with a time server on your network.

In all cases, you should verify that the Turbonomic clock is properly synchronized. To check the system clock:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Username: [your_private_password]

2. Verify your time settings.

Execute the date command. You should see results similar to:

```
Thu Feb 2 14:25:45 UTC 2019
```

To verify the time, you can execute the command, `timedatectl`. The output should be similar to:

```
Local time: Fri 2019-12-06 21:09:26 UTC
Universal time: Fri 2019-12-06 21:09:26 UTC
RTC time: Fri 2019-12-06 21:09:27
Time zone: UTC (UTC, +0000)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

This tells you whether you have NTP enabled, and whether it is currently synchronized, along with other time synchronization information.

If the output is correct *and* your environment has access to the internet, you can assume the system clock is synchronized.

If the output is incorrect, or if you need to configure synchronization with a time server on your network, you must configure `chrony` on the server instance.

To set up `chrony` on your Turbonomic instance:

1. Open an SSH terminal session to your Turbonomic instance.
2. Open the `chrony` configuration file.

For example, execute the command: `sudo vi /etc/chrony.conf`

3. Specify the time servers that you want to use in your environment.

The `chrony` file includes the following statements to configure time servers:

```
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst
```

Enter statements for the servers you want to use. Then delete or comment out the statements that you do not want to use.

Specify a time server via the following command syntax:

```
server My_Time_Server_Name iburst
```

4. Save the file.
5. Restart the `chrony` service.

Execute the command: `sudo systemctl restart chronyd`

6. Verify that your time is correct.

Execute the `date` command. You should see results similar to:

```
Fri Dec 6 21:09:26 UTC 2019
```

To verify the time has been synchronized, you can execute the command, `timedatectl`. The output should be similar to:

```
Local time: Fri 2019-12-06 21:09:26 UTC
Universal time: Fri 2019-12-06 21:09:26 UTC
RTC time: Fri 2019-12-06 21:09:27
Time zone: UTC (UTC, +0000)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: n/a
```

To verify the time, compare the `date` output with the output from a known UTC time server.

If the output is correct you can assume the system clock is synchronized.

If the output is incorrect, contact your support representative.

(Important) Verifying your MariaDB Version

For its default historical database, Turbonomic currently supports MariaDB version 10.5.20. This support includes comprehensive testing and quality control for Turbonomic usage of the historical database.

If you are running Turbonomic installed as a VM image (OVA or VHD), and using the database that is included in that image installation, then you must use version 10.5.20. If you are updating your version of

Turbonomic (instead of installing it for the first time), then you should make sure you are using the correct version of MariaDB with your installation.

This section shows you how to check the version of MariaDB on your VM image installation of Turbonomic. Also, if you have used the update script to updated your Turbonomic to version 8.3.5 or later, you can use the steps in this section to update your MariaDB.

Important:

It is a requirement that you run MariaDB version 10.5.20 or later. Turbonomic can operate with other versions of MariaDB; however, it is fully tested to operate with MariaDB version 10.5.20.

Important:

Because of a known issue, *you must never use* MariaDB versions 10.5.14, 10.5.15, 10.6.7, 10.7.3, or 10.8.2.

Turbonomic also supports MySQL 5.7.x, deployed as a custom installation.

When you initially installed Turbonomic, that installation included MariaDB running a specific version. As you update your Turbonomic version, the MariaDB version remains the same. The first release of Turbonomic that included MariaDB 10.5.20 is 8.7.6. If you initially installed an earlier version, and you have not explicitly updated your MariaDB to 10.5.20, then you must do it now.

For VM image installations, it is possible to configure the installation to use a remote database (external to the VM). For such deployments, you must manage the database versioning yourself. If you are using a remote MariaDB instance, we recommend that you use version 10.5.20. For a remote MySQL, you should use version 5.7.x.

For installations on a Kubernetes cluster (not deployed as a Turbonomic VM image), if you are using MariaDB we recommend that you use version 10.5.20; however, MariaDB 10.2 for Azure DB Services is tolerated. You can find a download package at: <https://archive.mariadb.org/mariadb-10.5.20/yum/centos7-amd64>. For MySQL, you should use version 5.7.x. For such deployments, you must manage the database versioning yourself.

Checking your MariaDB Version

To check the version of MariaDB running on your Turbonomic OVA:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Username: [your_private_password]

2. Check the MariaDB version.

```
mysql -u root --password=my_pwd -e "SHOW VARIABLES LIKE 'version';"
```

The output should be similar to:

```
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| version       | 10.5.20-MariaDB    |
+-----+-----+
```

If the version is lower than 10.5.20-MariaDB, then you must update your database.

If your version is equal to or higher than 10.5.20-MariaDB you should not perform the update steps below.

Updating your MariaDB

If you are using Turbonomic installed as a VM image, and you are using the default MariaDB that was installed with that image, you must run MariaDB version 10.5.20.

Important:

Before you perform these steps, you must have already used the update script to update your Turbonomic to version 8.3.5 or later.

Using the update script to update to 8.3.5 or later installs the necessary MariaDB update scripts. If your current version of Turbonomic is lower than 8.3.5, then update your installed version before executing these steps. For instructions to use the product update script, see [Updating Turbonomic to a New Version](#).

If your MariaDB version is equal to or higher than 10.5.20-MariaDB you should not perform these update steps.

If your current Turbonomic version is earlier than 8.3.5 and you want to update MariaDB but you do *not* want to update Turbonomic, contact your support representative.

To update your MariaDB on your Turbonomic VM:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: [your_private_password]

2. If you perform offline updates, ensure the VM is mounted on the Turbonomic update ISO image.

Note:

If you do not allow internet access, then you must perform offline updates of Turbonomic. When you complete the update, the system automatically unmounts the ISO image. To perform the MariaDB update, your Turbonomic instance **must be mounted** on the same ISO image that you used to update it to version 8.3.5 or later.

If you perform an offline update and mount the ISO image, you can skip step 3 below since the mariadbUpgrade.sh is on the ISO image.

For information about offline updates and mounting the ISO image, see [Offline Update](#).

3. (Online updates only) Set up the MariaDB update script.

- a. Log in to the Turbonomic VM.

Use SSH to log in to the Turbonomic VM using the turbo account and password.

- b. Change to the scripts directory.

```
cd /opt/local/bin
```

- c. Get the latest version of the script.

```
curl -O --proxy PROXY_NAME_IP:PORT \  
https://\  
raw.githubusercontent.com/turbonomic/t8c-install/master/bin/mariadbUpgrade.sh
```

Where `--proxy PROXY_NAME_IP:PORT` is an optional specification to execute the download through a proxy.

Note:

If you perform offline updates, download the script to your local machine.

- d. Ensure the script is saved on the Turbonomic server at the location:

```
/opt/local/bin/mariadbUpgrade.sh.
```

Note:

If you perform offline updates, use a secure shell session to upload the script from your local machine, to the Turbonomic server.

4. Execute the MariaDB update script.

Before you execute the script, you will need to know the MariaDB password. By default, this password is `vmturbo`.

a. Make the script executable.

Note:

If you perform offline updates and have already run the offline upgrade script, the updated `mariadbUpgrade.sh` is copied from the ISO image to the `/opt/local/bin` directory with the executable flag enabled. You can skip this step and execute the script as noted below.

```
sudo chmod +x /opt/local/bin/mariadbUpgrade.sh
```

b. Execute the database update script:

```
sudo /opt/local/bin/mariadbUpgrade.sh
```

The script updates the version of MariaDB. It also increases size limits for the allowed packets, and buffer and log sizes for the innodb. The script output should include the following (where `Total Memory` and `buffer pool size` can vary depending on your VM configuration):

```
=====
Update the mariadb configuration
=====
Total Memory: 128773 MB
Changing Innodb buffer pool size to: 9216 MB
Changing max allowed packets to: 1G
Changing innodb log file size to: 10G
=====
```

5. Verify the updated MariaDB version.

When the script completes, you should be running version 10.5.20. To check the version, execute the following command:

```
mysql -u root --password=my_pwd -e "SHOW VARIABLES LIKE 'version';"
```

The output should be:

```
+-----+-----+
| Variable_name | Value                |
+-----+-----+
| version       | 10.5.20-MariaDB    |
+-----+-----+
```

6. Scale up the Turbonomic platform's pods.

To update the database, the script scales down your platform pods. When it completes, the script displays the following prompt:

```
#####
When confirmed the mariadb has been upgraded and is properly working, run:
kubectl scale deployment --replicas=1 t8c-operator -n turbonomic
#####
```

After you verify that the correct version of MariaDB is running, scale up the platform:

```
kubectl scale deployment --replicas=1 t8c-operator -n turbonomic
```


(Important) Increasing your Database Capacities

For Turbonomic versions 8.0.6 or later, your historical database must provide certain storage size capacities. For MariaDB or MySQL installations, you must ensure your database provides the necessary messaging and logging capacity. (Turbonomic supports MariaDB version 10.5.20 or MySQL 5.7.X for the historical database. For more information, see [Verifying your MariaDB Version](#).)

Note:

If you installed Turbonomic as a VM image (OVA or VHD), the image includes MariaDB for the historic database. Most installations use that instance. However, you can use an external database if you like. In addition, you can install Turbonomic directly to a Kubernetes cluster (instead of installing the VM image), and you can provide your own historical database.

Do I Need to Increase My DB Capacities?

The way you have deployed the historical database in your Turbonomic installation determines whether you must perform this procedure:

For this installation	DB storage update requirements
<ul style="list-style-type: none"> You installed Turbonomic as a VM image at version 8.0.6 or earlier You use the default MariaDB as your historical database You updated Turbonomic to version 8.0.7 or later You ran the script that updates your MariaDB version (see Verifying your MariaDB Version). 	<p>You do not need to increase your DB capacities. You can skip this procedure.</p> <p>Note: If you are still running MariaDB 10.5.6 or 10.5.9, you must update to 10.5.20. See Verifying your MariaDB Version.</p>
<ul style="list-style-type: none"> You installed Turbonomic as a VM image at version 8.0.6 or earlier You use the default MariaDB as your historical database You have not updated Turbonomic to version 8.0.7 or later You have not run the script that updates your MariaDB version (see Verifying your MariaDB Version). 	<p>You must increase your DB capacities.</p> <p>Note: The simplest way to increase capacities and ensure you are running the correct version of MariaDB is to update Turbonomic to version 8.2.3 or later (see Updating Turbonomic to a New Version). If you want to update your MariaDB without updating Turbonomic, contact your support representative.</p>
<ul style="list-style-type: none"> You installed Turbonomic as a VM image You use an <i>external</i> MariaDB or MySQL as your historical database 	<p>You must increase your DB capacities.</p>
<ul style="list-style-type: none"> You installed Turbonomic directly on a Kubernetes cluster (not deployed as the Turbonomic VM image) You use MariaDB or MySQL as your historical database 	<p>You must increase your DB capacities.</p>

Checking you Current DB Capacities

Before you update your DB capacities, check to make sure they need to be increased. Open an SSH terminal session that can administer your database, and execute the following commands:

- Check the size for `max_allowed_packet`.

```
mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'max_allowed_packet';"
```

You should see output similar to:

```
+-----+-----+
| max_allowed_packet | {PacketSize} |
+-----+-----+
```

Where **{PacketSize}** is your current value. If it is greater than or equal to 1073741824, you do not need to increase this capacity.

- Check the size for `innodb_log_file_size`.

```
mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'innodb_log_file_size';"
```

You should see output similar to:

```
+-----+-----+
| innodb_log_file_size | {LogSize} |
+-----+-----+
```

Where **{LogSize}** is your current value. If it is greater than or equal to 10737418240 (10 times the max allowed packet size), you do not need to increase this capacity.

Configuring the Capacities

To configure your DB capacities:

1. Edit your `server.cnf` file to set the following server system variables:

- `max_allowed_packet = 1G`
- `innodb_log_file_size = 10G`

To make these settings, open the `server.cnf` file in an editor, change the values for these system variables, and then save your changes.

By default, you can find the configuration file at this location:

```
/etc/my.cnf.d/server.cnf
```

2. Restart your DB instance.

Execute the command:

- For MariaDB:

```
sudo systemctl restart mariadb
```

- For MySQL:

```
sudo systemctl restart mysqld
```

3. Verify the system capacities.

After the database restarts, execute the following commands:

- `mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'max_allowed_packet';"`

You should see output similar to:

```
+-----+-----+
| Variable_name      | Value      |
+-----+-----+
```

```
| max_allowed_packet | 1073741824 |
+-----+-----+
```

- `mysql -u root --password=vmturbo -e "SHOW VARIABLES LIKE 'innodb_log_file_size';"`

You should see output similar to:

```
+-----+-----+
| Variable_name      | Value          |
+-----+-----+
| innodb_log_file_size | 10737418240   |
+-----+-----+
```

(Optional) Increasing Available Disk Space

A standard installation of Turbonomic on a VM image includes a MariaDB database server for historical data. If you enable Embedded Reports, the platform also uses TimescaleDB Postgres database to manage the reports data. For various reasons, you might find that the default storage capacity for your database services is not sufficient. In that case, you need to increase the available storage capacity.

A common reason to increase this capacity is to accommodate estimated needs for Embedded Reports. The storage requirements for Embedded Reports can change over time as your environment changes, or as you increase the number of targets you configure your Turbonomic installation. For information about estimating Embedded Reports requirements, see [Embedded Reports Storage Requirement Estimates](#).

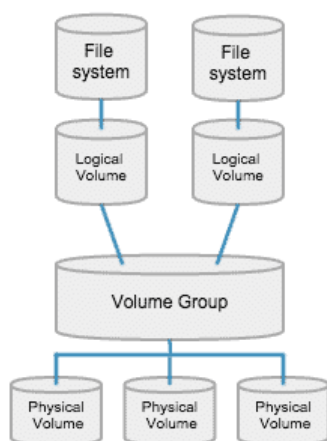
A summary of the steps you will perform is:

- Add a new disk to the VM
- Rescan the scsi devices
- Create a new LVM partition
- Create a physical volume (pv)
- Add the pv to the existing volume group (vg)
- Extend the logical volume (lv)
- Extend the file system to use the new lv
- To increase storage for Embedded Reports, increase the XFS quota

To increase space for MariaDB, you do not need to perform this step.

Logical Volume Management for Turbonomic Storage

The platform uses Logical Volume Management (LVM) to manage the VM disks. To increase database storage, you should add a new disk to the VM, and then use it to extend the LVM logical volume, `/dev/turbo/var_lib_mysql`. This logical volume serves both the historical database and the Embedded Reports database.



Increasing Storage – Procedure

To increase the storage space available to your databases:

1. Add a new disk to the VM.

Use the steps for your VM datacenter to add a new disk to the VM. Turbonomic installs as a VMware or a Hyper-V VM. Refer to the documentation for your hypervisor for the steps to add a new disk.

2. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: [your_private_password]

3. Rescan the scsi devices.

To make sure the new disk is available, rescan the scsi devices and then list your block devices.

To scan the devices, execute:

```
echo "- - -" > /sys/class/scsi_disk//0\:0\:0\:0/device/rescan
```

To check for the new disk, execute:

```
lsblk
```

The new disk should appear with a name similar to /dev/sdc. If you don't see the new disk, try this alternative to force a rescan:

- Check the number of scsi host devices that are on your VM:

```
ls /sys/class/scsi_host
```

You should see a list of devices, such as host0, host1, host2... hostn

- Scan each device

For each device execute the command (where **<hostn>** is a numbered host device such as host0 or host1):

```
echo "- - -" > /sys/class/scsi_host/host0/scan
```

- List the block devices

Execute `lsblk` again to list the block devices.

4. Create a new LVM partition.

Assuming the new disk is named `/dev/sdc1`, execute the command:

```
cfdisk /dev/sdc1
```

Then execute the operations:

- `new`
- `primary`
- `confirm size`
- `change type to 8E`
- `write`
- `quit`

5. Create the Physical Volume (pv).

Assuming the new disk is named `/dev/sdc1`, execute the command:

```
pvcreate /dev/sdc1
```

6. Add the new pv to the existing Volume Group.

Assuming the new disk is named `/dev/sdc1`, execute the command:

```
vgextend /dev/turbo /dev/sdc1
```

7. Extend the Logical Volume (lv) to use the free space in the new pv.

First list the physical extents (PE) that are available. Execute the command:

```
vgdisplay
```

You should see results similar to:

```
Free PE / Size      128000 / 500.00 GiB
```

In this example, 128000 is the amount to extend the lv. For this example, execute the command:

```
lvextend -l +128000 /dev/turbo/var_lib_mysql
```

8. Extend the XFS file system to use all the current lv space.

Before you extend the XFS, view the free disk space and record the number. To verify that you have increased the available space, you will compare this value to the free space after you have extended XFS. Execute the command:

```
df -h
```

Then extend the XFS capacity:

```
xfs_growfs /dev/turbo/var_lib_mysql
```

Then list the updated free disk space and compare it to your original number:

```
df -h
```

9. If you are increasing capacity for Embedded Reports, extend the XFS quota for the TimescaleDB.

To increase space for MariaDB, you do not need to perform this step.

To increase capacity for the Timescale DB, you need to increase the quota for that process by the amount you want. The quota name is `Postgresql`.

For example, assume you added a 400 GB volume, and the current `Postgresql` quota is 400 GB. In that case, you could increase the quota to 800 GB. Following this example, execute the command:

```
xfs_quota -x -c 'limit -p bhard=800g Postgresql' /var/lib/dbs
```

To see the current quotas set for `/var/lib/dbs`, execute the command:

```
xfs_quota -xc 'report -pbih' /var/lib/dbs
```

(Optional) Enforcing Secure Access via LDAP

If your company policy requires secure access, you can use a certificate with your LDAP service to set up secure access for your users. For example, you can configure Active Directory (AD) accounts to manage *External Authentication* for users or user groups. The user interface to enable AD includes a **Secure** option, which enforces certificate-based security. For more information, see [Managing User Accounts](#).

If your LDAP service uses a Certificate Authority (CA), then the certificate signed by that CA should support this feature as it is. Simply turn on the **Secure** option when you are setting up your AD connection.

If your LDAP service uses a self-signed certificate, then you must install that certificate on the Turbonomic authorization pod. The steps you will perform include:

- Get the certificate from your LDAP server
- Import the certificate to the platform's TrustStore
- Add the certificate to the Turbonomic platform's authorization pod
- Enable the TrustStore in the Turbonomic platform's Operator chart

The procedure to install the certificate is different depending on the version of Turbonomic that you initially installed, or whether you are using Red Hat OpenShift.

- If you *initially installed* Turbonomic 8.1.5 or later, see [8.1.5 or Later OVA: Installing a Self-signed Certificate](#)
- If you *initially installed* Turbonomic 8.1.4 or earlier, see [8.1.4 or Earlier OVA: Installing a Self-signed Certificate](#).
- If you installed Turbonomic using Red Hat OpenShift, see [Red Hat OpenShift: Installing a Self-signed Certificate](#).

Note:

To determine when your Turbonomic instance was initially installed, you can execute a `describe` on the auth pod to see if the instance is using gluster. Execute the following command:

```
kubectl describe pod auth <TAB>
```

If you see the following line, continue using the [8.1.5 or Later OVA: Installing a Self-signed Certificate](#) steps:

```
auth-secret: Type: Secret (a volume populated by a Secret) SecretName: auth-secret Optional: true
```

If you do not see that line, use the [8.1.4 or Earlier OVA: Installing a Self-signed Certificate](#) steps.

8.1.5 or Later OVA: Installing a Self-signed Certificate

This section describes how to set up secure access from an LDAP server, to a Turbonomic platform that was *initially installed* as version 8.1.5 or later. It assumes you have authorization to get a certificate from the LDAP server, as well as admin authority on the Turbonomic platform.

To set up secure access:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: [your_private_password]

2. Download your LDAP Server certificate to the Turbonomic instance.

Acquire a certificate from your LDAP administrator, and download it to the Turbonomic platform. For example, you can download it to the file `/tmp/ldapserver.crt`:

3. Import the `.crt` file to the Turbonomic TrustStore.

This step modifies the `cacerts` file on the Turbonomic platform.

Note:

To import a certificate to the Turbonomic TrustStore, you must use the `keytool` utility. To install this utility, execute the command:

```
sudo yum install java-1.8.0-openjdk
```

This installs the utility in `/usr/bin/keytool`.

If an alias for an LDAP certificate already exists, delete that certificate. For example, assuming the alias `ldapcert1`, execute the following command:

```
keytool -delete -alias ldapcert1 -keystore cacerts -storepass changeit
```

Then use the following command to import your new certificate to the TrustStore:

```
keytool -import -alias ldapcert1 -file /tmp/ldapserver.crt -keystore cacerts \
  -deststoretype jks -storepass changeit -noprompt
```

4. Create an auth secret from the `cacerts` file.

```
base64 cacerts > auth-secrets.yaml
```

5. Open the secrets file for editing.

```
vi auth-secrets.yaml
```

6. Edit the file to make it a valid yaml file.

- a. Indent every line of the certificate by four spaces.

When you created the file, you concatenated the contents of the certificate. The first step is to indent the certificate by four spaces. For example, in a `vi` editor, execute the following command:

```
:%s/^/    /g
```

- b. Add data fields to the secrets file.

Add the following text to the top of the file:

```
apiVersion: v1
kind: Secret
metadata:
  name: auth-secret
data:
  cacerts: |
```

- c. Save your changes.

The completed file should be similar to:

```
apiVersion: v1
kind: Secret
metadata:
  name: auth-secret
data:
```

```

cacerts: |
/u3+7QAAAAIAAAAABAAAAAGAFY2VydDEAAAF5H21EigAFWC41MDkAAAYQMIIGDDCCBPSgAwIBAgIT
HAAAAARHIFJDLbG90sAAAAABETANBgbkqhkiG9w0BAQUFAADBcMRMwEQYKCZImiZPyLQGvBGRYDY29t
MRcwFQYKCZImiZPyLQGvBGRYHdm10dXJibzEUMBIGCgmSjomT8ixkARKkWBGNvcnAxFjAUBGNVBAMT
DWNvcnAtREVMTDEtQ0EwHhcNMjEwNDA4MDM00TEyWhcNMjIwNDA4MDM00TEyWjAhMR8wHQYDVQQD
ExZkZkXwMS5jb3JwLnZtdHVyYm8uY29tMIIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
sCXuh2MTtRFERYU/aKgdbgyjLezNuWf6nmZveZuHdaJDpflHJl1zhwfyYRTGfSSusVo4polJS4WqPZ
T3Zk8f2IaX04RpfQErq5N3uY/BxFkATWLMdiquSd0Di798k2diYXAxVzMMfmIkBBYJta9oztum
uXyh/42dX0GznQ5fFuxosgAksZ6CnXGDKrTB1b0bHpST1z1PdG+fJ+f9Tq7Iff0YdVbuedFTwsik
Z0JgDCIRrmmS0JphiHdBqJ6ZLdbSeEzBIbboiQs81pAELw7V0ZzUfKv6y8+zMTACGwpVPJfV7LX
RlW1TWcqhXVA0mroe2WcU8KJE6XZTBxp7z7dzwIDAQABo4IDADCCAvwwwLwYJKwYBBAAGCNxQCBCIE
IABEAG8AbQhAgkAbgBDAG8AbgB0AHIAbwBsAGwAZQByMB0GA1UdJjQvWMBQGCCsGAQUFBwMCBgggr
BgEFBQcDATAOBgNVHQ8BAf8EBAMCBaAweAYJKoZIhvcNAQkPBGswaTAOBggqhkiG9w0DAGICAIAw
DgYIKoZIhvcNAwQCAgCAMAsGCWCGSFA1AwQBKjALBg1ghkgBZQMEAS0wCwYJYIZIAWUDBAECMAAG
CWCGSFA1AwQBBTAAHBgUrdgMCBzAKBggqhkiG9w0DBzBCBgNVHREEOzA5oB8GCSsGAQQBgjczAaAS
BBDswj1Hut/nQZ0uK2aUglGbhZkZwXsMS5jb3JwLnZtdHVyYm8uY29tMB0GA1UdDgQWBRR6M7Hb
BiirpjIXQ3PXXScB8LkmRDAfBgNVHSMEGDAWgBRjs9l3e17SuKUDM1rHHRhBKENGaDCB0QYDVR0f
BIHJMIGHMIHDoIHaoIG9hoG6bGRhcDovLy9DTj1jb3JwLURFTExwLUNBLENOPWRlBwXLENOPUNE
UCxDTj1QdWJsaWMLMjBLZXk1MjBTZXJ2aWN1cyxDTj1TZXJ2aWN1cyxDTj1Db25maWd1cmF0aW9u
LERDPWNvcnAsREM9dm10dXJibyxEQz1jb20/Y2VydG1maWNhdGVSZXZvY2F0aW9uTG1zdD9iYXN1
P29iamVjdENsYXNzPWNSSTERpc3RyaWJ1dGlvb1BvaW50MIHBBgggrBgEFBQcBAQSBuJCBtZCBtAYI
KwYBBQUHMAKGGadsZGFwOi8vL0NOPWNvcnAtREVMTDEtQ0E0sQ049QU1BLENOPVB1YmXpYyUyMetl
eSUyMfN1cnZpY2VzLENOPVNi1cnZpY2VzLENOPUNvbmZpZ3VvYXRpb24sREM9Y29yY29yY29yY29y
cmJvLERDPWNvbt9jQUN1cnRmZm1jYXR1P2Jhc2U/b2JqZWNOQ2xhc3M9Y2VydG1maWNhdGlvbkF1
dGhvcml0eTANBgkqhkiG9w0BAQUFAAOCAQEAADP60YLONkZ2j6gaBdfdoIJtvn1g1qXTsRrtFuUcF
C9mUxL0G5Tudr0VlyEnLH2wtj10CGsIi54+aPGYiE1XijThEe1WTHa02hk1RLdNrM8KxUp3tUNb/
cP4d+eY2t97wVWgxp19MStiND8+7M2+65daoEu5IOLtq4LC7Y1CSXay19N5HdiGBHV5L07PTZ261
qDzShSb0ZwtG7++5VkvqeVEIfs3hUYdaItz0Zu6sym90aUcVn5wohV1GPPqGDvVCG5Kf50hsZfmy
1tNlaqiiqLMnYVMA93CkPFfjoP9gmGFJky0yTfh6G8HuqbI7guddDsUqMQTT3uv3EBwSYeIm0ya7
Zye5C4NnsAFnx8k0wXdsVERC

```

7. Apply this secrets file to the platform environment.

```
kubectl apply -f auth-secrets.yaml
```

8. Update the platform's Operator Chart to use the cacerts certificate that you created in the secrets file.

a. Open the chart file for editing.

Open the file, `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml`.

b. Add the certification secret as an authorization spec for the component options.

In the chart file, find the `spec :` section. Within that section, find the `auth :` subsection.

This should be the second subsection in `spec :`, after `global :`. If there is no `auth :` subsection, you can add it to `spec :`.

c. Add the certification secret to the file:

You will add the secret's path to a `javaComponentOptions :` statement within the `auth :` subsection. Add the path as a `-D` option. The `auth :` subsection should be similar to the following, with `auth` indented by two spaces and `javaComponentOptions` indented by four spaces:

```

# Pass in the JAVA_OPTS to the auth POD to set up additional options such as
# a trustStore for AD Certificate(s) for LDAPS (Secure LDAP)
auth:
  javaComponentOptions: "-Djavax.net.ssl.trustStore=/home/turbonomic/data/helper_dir/
cacerts"

```

d. Apply your Operator Chart changes to the Turbonomic platform.

Execute the following command:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

This restarts the authorization component so it can use the new setting.

8.1.4 or Earlier OVA: Installing a Self-signed Certificate

This section describes how to set up secure access from an LDAP server, to a Turbonomic platform that was *initially installed* before version 8.1.5. It assumes you have authorization to get a certificate from the LDAP server, as well as admin authority on the Turbonomic platform.

To set up secure access:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: [your_private_password]

2. Download your LDAP Server certificate to the Turbonomic instance.

Acquire a certificate from your LDAP administrator, and download it to the Turbonomic platform. For example, you can download it to the file `/tmp/ldapserver.pem`:

3. Import the `.pem` file to the Turbonomic TrustStore.

This step modifies the `cacerts` file on the Turbonomic platform.

Note:

To import a certificate to the Turbonomic TrustStore, you must use the `keytool` utility. To install this utility, execute the command:

```
sudo yum install java-1.8.0-openjdk
```

This installs the utility in `/usr/bin/keytool`.

If an alias for an LDAP certificate already exists, delete that certificate. For example, assuming the alias `ldapcert1`, execute the following command:

```
keytool -delete -alias ldapcert1 -keystore cacerts -storepass changeit
```

Then use the following command to import your new certificate to the TrustStore:

```
keytool -import -alias ldapcert1 -file /tmp/ldapserver.pem -keystore cacerts \
  -deststoretype jks -storepass changeit -noprompt
```

4. Add the TrustStore to the Turbonomic authorization pod.

Execute the following command to copy the `cacerts` file into the authorization pod:

```
kubectl cp cacerts $auth_pod:/home/turbonomic/data/cacerts
```

5. Update the platform's Operator Chart to use the TrustStore.

- a. Open the chart file for editing.

Open the file, `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml`.

- b. Add the TrustStore as an authorization spec for the component options.

In the chart file, find the `spec :` section. Within that section, find the `auth :` subsection.

This should be the second subsection in `spec :`, after `global :`. If there is no `auth :` subsection, you can add it to `spec :`.

- c. Add the TrustStore to the `auth :` subsection.

You will add the TrustStore path to a `javaComponentOptions :` statement within the `auth :` subsection. Add the path as a `-D` option. Use the same path that you copied the `cacerts` file to in the Turbonomic authorization pod. In the above example, you copied it to `$auth_pod:/home/turbonomic/data/cacerts`.

Following the above example, the `auth:` subsection should be similar to the following, with `auth` indented by two spaces and `javaComponentOptions` indented by four spaces:

```
# Pass in the JAVA_OPTS to the auth POD to set up additional options such as
# a trustStore for AD Certificate(s) for LDAPS (Secure LDAP)
auth:
  javaComponentOptions: "-Djavax.net.ssl.trustStore=/home/turbonomic/data/cacerts"
```

- d. Save the `charts_v1alpha1_x1_cr.yaml` file.
6. Apply your Operator Chart changes to the Turbonomic platform.

Execute the following command:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

This restarts the authorization component so it can use the new setting. As the component restarts, the `rsyslog` output should include the following Message:

```
-Djavax.net.ssl.trustStore=/home/turbonomic/data/cacerts
```

Red Hat OpenShift: Installing a Self-signed Certificate

This section describes how to set up secure access from an LDAP server, to a Turbonomic platform. It assumes you have authorization to get a certificate from the LDAP server, as well as admin authority on the Turbonomic platform.

To set up secure access:

1. Download your LDAP Server certificate to the Turbonomic instance.

Acquire a certificate from your LDAP administrator, and download it to the Turbonomic platform. For example, you can download it temporarily to the file `/tmp/ldapserver.crt`.

2. Import the `.crt` file to the Turbonomic TrustStore.

This step modifies the `cacerts` file on the Turbonomic platform.

Note:

To import a certificate to the Turbonomic TrustStore, you must use the `keytool` utility. To install this utility, execute the command `yum install` or `brew install`, depending on your operating system. For example:

```
sudo yum install java-1.8.0-openjdk
```

This installs the utility in `/usr/bin/keytool`.

If an alias for an LDAP certificate already exists, delete that certificate. For example, assuming the alias `ldapcert1`, execute the following command:

```
keytool -delete -alias ldapcert1 -keystore cacerts -storepass changeit
```

Then use the following command to import your new certificate to the TrustStore:

```
keytool -import -alias ldapcert1 -file /tmp/ldapserver.crt -keystore cacerts -deststoretype jks -storepass changeit -noprompt
```

3. Add the TrustStore to the Turbonomic authorization pod.

Execute the following command to copy the `cacerts` file into the authorization pod:

```
oc cp cacerts $auth_pod:/home/turbonomic/data/cacerts
```

4. Update the platform's Operator Chart to use the TrustStore.

- a. Open the yaml file for editing.

```
oc edit xl xl-release
```

Alternatively, you can edit the file directly in the Red Hat OpenShift UI.

- b. Add the TrustStore as an authorization spec for the component options.

In the yaml file, find the `spec :` section. Within that section, find the `auth :` subsection.

This should be the second subsection in `spec :`, after `global :`. If there is no `auth :` subsection, you can add it to `spec :`.

- c. Add the TrustStore to the `auth :` subsection.

You will add the TrustStore path to a `javaComponentOptions :` statement within the `auth :` subsection. Add the path as a `-D` option. Use the same path that you copied the `cacerts` file to in the Turbonomic authorization pod.

The `auth :` subsection should be similar to the following, with `auth` indented by two spaces and `javaComponentOptions` indented by four spaces:

```
# Pass in the JAVA_OPTS to the auth POD to set up additional options such as
# a trustStore for AD Certificate(s) for LDAPS (Secure LDAP)
spec:
  auth:
    javaComponentOptions: "-Djavax.net.ssl.trustStore=/home/turbonomic/data/cacerts"
```

This restarts the authorization component so it can use the new setting. As the component restarts, the `rsyslog` output should include the following message or you can `exec` into `auth` pod and execute `ps -ef | grep java`.

```
-Djavax.net.ssl.trustStore=/home/turbonomic/data/cacerts
```

(Optional) Adding a Certificate for Securing the Turbonomic UI

If your company policy requires SSL connections via trusted certificate, Turbonomic enables you to install a trusted certificate from a known certificate authority.

Requesting a Certificate

The first step is to acquire a certificate. The following steps describe how to generate a certificate request.

1. Open a shell terminal session.

Open an SSH terminal session on your Turbonomic instance. Log in as `turbo`, and use the password that you created for the administration account in the installation steps above. For information, see the installation step, [Set up the Turbonomic System Administrator account](#).

2. Change to the directory where you want to store the private key file.

If your shell session is on your Turbonomic instance, you should use the `/opt/turbonomic` directory:

```
cd /opt/turbonomic
```

3. Create and save the private key file.

Execute the command to create a private key file.

For this example, the private key file is named `myPrivate.key`

```
openssl genrsa -out myPrivate.key 2048
```

You will need this file later. If you are in a session on your Turbonomic instance, you might want to copy the file to your local machine.

4. Create a file to contain the information that will generate the signed certificate request (CSR).

```
vi certsignreq.cfg
```

5. Add the request data to the `certsignreq.cfg` file.

In the file, insert the following code. For any fields marked by angle brackets (for example `<city>`), provide the indicated value. For example, your country, city, company, etc.

```
[req]
ts = 2048
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn

[dn]
C=<country, 2 letter code>
L=<city>
O=<company>
OU=<organizational unit name>
CN=<FQDN>
emailAddress=<email address>

[req_ext]
subjectAltName = @alt_names

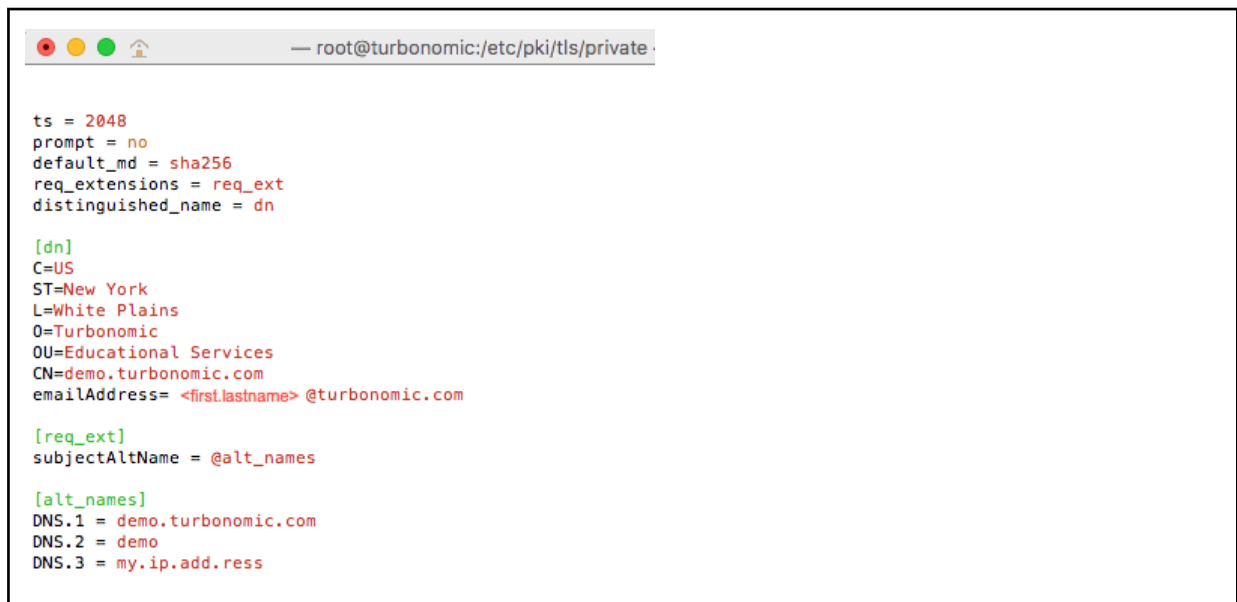
[alt_names]
DNS.1 = <FQDN>
DNS.2 = <server's short name>
DNS.3 = <server's IP address>
```

Note:

For the CN field, specify the fully-qualified domain name of the Turbonomic instance.

Alternate names are other ways to access the Turbonomic instance. In the `[alt_names]` section, the value for the `DNS.1` field is required. For `DNS.1`, specify the fully-qualified domain name of the Turbonomic instance. Values for the `DNS.2` and `DNS.3` are optional. You can add more `DNS.<n>` fields if needed.

For example:



```
root@turbonomic:/etc/pki/tls/private:
ts = 2048
prompt = no
default_md = sha256
req_extensions = req_ext
distinguished_name = dn

[dn]
C=US
ST=New York
L=White Plains
O=Turbonomic
OU=Educational Services
CN=demo.turbonomic.com
emailAddress= <first.lastname> @turbonomic.com

[req_ext]
subjectAltName = @alt_names

[alt_names]
DNS.1 = demo.turbonomic.com
DNS.2 = demo
DNS.3 = my.ip.add.ress
```

6. Write and quit the file.

Press **esc**, type `:wq!`, and press **Enter**.

7. Generate the certificate request file.

In this example, we name the file `myRequest.csr`.

Execute the command:

```
openssl req -new -sha256 -nodes -out myRequest.csr -key \
myPrivate.key -config certsignreq.cfg
```

8. Send the generated request file to your certificate authority.

If you generated the file on your Turbonomic instance, you should transfer the file to your local machine. The path to the certificate request file on your remote machine is `/opt/turbonomic/myRequest.csr`.

Your certificate authority will use this file to create the certificate for you.

If your certificate authority gives you an encoding choice between DER and Base 64, choose **Base 64**.

9. When you receive the certificate, save it to disk.

If you did not receive the certificate encoded in Base 64, you must convert it from DER to Base 64. Execute the following command, assuming the certificate is named `MyCertificate.crt`:

```
openssl x509 -inform der -in MyCertificate.der -out MyCertificate.crt
```

Installing the Signed Certificate in Turbonomic

Once you have obtained the signed certificate, you can install it on your Turbonomic instance. You will use the private key and certificate files you obtained when requesting the signed certificate:

- `myPrivate.key`
- `MyCertificate.crt`

To install the signed certificate:

1. Open an SSH terminal session on your Turbonomic instance.
2. Add the key and certificate data to your Turbonomic `charts.yaml` file.

Open the file: `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml`

Find the section for `global` parameters. Under the `global` parameters, create the `ingress:secrets` section, and then create entries for `certificate`, `key`, and `name`.

Your global parameters should be similar to the following:

```
global:
  ingress:
    secrets:
      - certificate: |
          -----BEGIN CERTIFICATE-----
          SAMPLE PUBLIC KEY
          -----END CERTIFICATE-----
        key: |
          -----BEGIN RSA PRIVATE KEY-----
          SAMPLE PRIVATE KEY
          -----END RSA PRIVATE KEY-----
        name: nginx-ingressgateway-certs
```

For the fields you added:

- `certificate`: This field holds the content of your `MyCertificate.crt` file. Open that file to copy its contents and paste them here.
 - `key`: This field holds the content of your `myPrivate.key` file. Open that file to copy its contents and paste them here.
 - `name`: This field is required, and the name must be `nginx-ingressgateway-certs`.
3. Apply the changes you made to the CR file.

Execute the command:

```
kubectl apply -f \
  kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

4. Restart the nginx pod.

To require a certificate for HTTPS access, you must restart the nginx pod:

a. Get the full name of the pod.

Execute the command `kubectl get pods -n turbonomic`. In the output, look for the entry for nginx. You should find an entry similar to:

```
nginx-5b775f498-sm2mm          1/1    Running    0
```

b. Restart the pod.

Execute the following command, where **<UID>** is the generated ID for the pod instance:

```
kubectl delete pod nginx-<UID>
```

This should restart the nginx pod. After restart, Turbonomic will then require a certificate for HTTPS access.

(Optional) Adding Additional CA Certificates for Probes

If your targets require SSL connections via trusted certificate, Turbonomic enables you to install a trusted certificate on the associated probe component.

The Turbonomic platform includes a number of probe components that it uses to connect to targets and discover their data. This procedure assumes setup for one component, the *Dynatrace* probe. You can use the same steps for other probes, providing a different Kubernetes Secret Name for each.

To install a certificate on a probe component, you must know the Kubernetes secret name for the given probe. This table lists the probes that you can configure, plus their secret names. If you must configure secure access for any probes not listed here, contact your support representative.

Probe	K8s Secret Name
mediation-appinsights	appinsights
mediation-aws mediation-awsbilling mediation-awscost	aws
mediation-azure mediation-azurecost mediation-azuresp mediation-azurevolumes	azure
mediation-azurebilling	azurebilling
mediation-azureea	azureea
mediation-dynatrace	dynatrace
mediation-newrelic	newrelic

Installing the Signed Certificate on the Probe Component

This procedure assumes you already have a valid `.crt` file. If you do not have the certificate file, ask your networking team to generate one for you.

Once you have obtained the signed certificate, you can install it on your probe instance. You will use the certificate file you obtained:

```
MyCertificate.crt
```

To install the signed certificate on a probe:

1. Copy the certificate from your local machine to the Turbonomic instance.
Use SCP to copy the `MyCertificate.crt` from your local machine to the `/tmp` directory on the instance.
2. Open an SSH terminal session on your Turbonomic instance, using the `turbo` user account.
3. Obtain the trust store from the probe component.

First, get the ID for the pod that runs the probe. To get the ID, execute the command:

```
kubectl get pods
```

This lists the pods running in the Turbonomic platform, including their IDs. Record the ID of the pod you want to configure.

To get the CA trust store, execute the following command, where **<Probe-Pod-Id>** is the ID you recorded:

```
kubectl cp <Probe-Pod-Id>:etc/pki/ca-trust/extracted/java/cacerts cacerts
```

4. Import the certificate into the pod's keystore.

As part of this step, you will ensure that the certificate is in Base64 format and you will create a yml file using the K8s Secret Name for the probe. While still in the bash session, execute the following commands:

- `chmod 775 cacerts`
- ```
keytool -import -alias probe_certificate -file \
MyCertificate.crt -keystore cacerts -deststoretype jks \
-storepass changeit -no-prompt
```

Where **MyCertificate.crt** is the name of the certificate that you acquired.

- `base64 cacerts > <Secret_Name>-secrets.yaml`

Where **<Secret\_Name>-secrets.yaml** is the yml file you will create using the K8s Secret Name for the probe.

For example, assume you are enabling SSL for the Dynatrace probe. In that case, the secret name is `dynatrace`, and you would create the yml file:

```
dynatrace-secrets.yaml
```

5. Update the `<Secret_Name>-secrets` yml file you just created.

- a. While still in the bash session on the Turbonomic server, open the yml file created in the previous step in a vi editor session:

```
vi <Secret_Name>-secrets.yaml
```

- b. Align the base64 data to the yml format.

Type : to enter the command mode. For the command, type the following, where the whitespace token is four space characters:

```
:%s/^/ /g
```

Press RETURN to execute the command. Then save and exit the vi editor.

c. Then add the following content to the file above your Base64 data:

```
apiVersion: v1
kind: Secret
metadata:
 name: <Secret_Name>
data:
 cacerts: |
 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Your Base64 data should be in the cacerts section in place of the xxx characters in the example above.

6. Apply the yaml file to the Turbonomic platform.

Execute the following commands, where **<Probe-Pod-Id>** is the ID you recorded, and :

- `kubectl apply -f <Secret_Name>-secrets.yaml`

7. For each probe that you configure with a SSL certificate, add an entry in the chart\_v1alpha1\_cl\_cr.yaml file.

a. With a shell session running on the Turbonomic platform, open the following file in a text editor:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

b. Search the file for the entry for the probe that you are configuring. Use the probe names listed in the table above. For example, if you are configuring the Dynatrace probe, find the entry for mediation-dynatrace. If the entry does not exist in the file, you can add it to the spec : section at the same level as global : , with the probe entry indented by two spaces and javaComponentOptions indented by four spaces.

c. Underneath the probe entry, add the following entry for javaComponentOptions:

```
javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts
```

For example, if you are configuring the Dynatrace probe, the entry should be similar to:

```
mediation-dynatrace:
 javaComponentOptions: -Djavax.net.ssl.trustStore=/etc/targets/cacerts
 resources:
 limits:
 memory: 2Gi
```

d. Save and exit the chart\_v1alpha1\_cl\_cr.yaml file.

e. Apply the changed file to your Turbonomic platform.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/
charts_v1alpha1_x1_cr.yaml
```

## (Optional) Modifying the Certificates for Cluster Manager

For installations behind a firewall, to upload diagnostics from the clustermgr component, you must modify its certificates. If you decide to modify these certificates, you must be running Turbonomic version 8.1.5 or later.



These steps to modify the certificates on `clustermgr` assume you have already generated the certificates that you want to add to the cluster manager.

1. Open an SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: `turbo`
- Username: `[your_private_password]`

2. Get the full name of the `clustermgr` pod.

Execute the command:

```
kubectl get pods -n turbonomic | grep clustermgr
```

The result should be similar to:

```
clustermgr-5f487f58f-tf84b 0/1 Running 52 2d4h
```

In this example, `clustermgr-5f487f58f-tf84b` is the full name of the pod, and `5f487f58f-tf84b` is the `POD_ID`.

3. Save a copy of the pod's current `ca-bundle.crt` file to `/tmp`.

Execute the following command, where `<POD_ID>` is the ID you get from the pod's full name.

```
kubectl cp \
clustermgr-<POD_ID>:etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem \
/tmp/ca-bundle.crt
```

4. Add your certificates to the bundle.

Repeat this command for each certificate, where `<MY_CERT>` is your certificate file.

```
cat <MY_CERT> >> /tmp/ca-bundle.crt
```

5. Create a Kubernetes secret for the modified certificates.

```
kubectl create secret generic clustermgr-secret --from-file=/tmp/ca-bundle.crt
```

6. Open the `cr.yaml` file for editing.

For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

7. Modify the `cr.yaml` file to use this secret.

Add the following to the file:

```
clustermgr:
 env:
 - name: component_type
 value: clustermgr
 - name: instance_id
 valueFrom:
 fieldRef:
 fieldPath: metadata.name
 - name: instance_ip
 valueFrom:
 fieldRef:
 fieldPath: status.podIP
 - name: serverHttpPort
 value: "8080"
 - name: kafkaServers
 value: kafka:9092
 - name: kafkaNamespace
 valueFrom:
 fieldRef:
 apiVersion: v1
 fieldPath: metadata.namespace
```

```
- name: CURL_CA_BUNDLE
 value: /home/turbonomic/data/ca-bundle.crt
```

8. Save your changes and apply the cr.yaml file.

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

If you watch the log with `grep ^clustermgr`, then you should see the appropriate curl command execute whenever you send diagnostics.

You can also check the .crt file in the cluster manager pod via the following commands, where <POD\_ID> is the ID you got from the pod's full name:

```
kubectl exec -it clustermgr-<POD_ID> bash
vi /home/turbonomic/data/ca-bundle.crt
```

## (Optional) Enabling Embedded Reports

The Turbonomic platform includes an Embedded Reporting component that you can choose to enable when you install the platform. Use Embedded Reporting to understand application resource management trends, and to share insights with stakeholders through reports and dashboards.

Embedded Reporting runs as its own component, as part of the Turbonomic platform. This architecture enhances performance and reduces storage requirements. It stores a history of your managed environment and then presents selective snapshots of this history through a set of standard dashboards and reports. You can create your own dashboards and reports to focus on other areas of concern.

### Note:

Embedded Reports use the TimescaleDB service, a time-series SQL database that provides fast analytics and scalability on a storage engine. These instructions are for the default configuration of TimescaleDB. As an alternative, you can configure an external deployment of TimescaleDB, and use that to manage your Embedded Reports data. For more information, see [Configuring an External TimescaleDB](#).

The method that you use to enable embedded reports depends on the version status of your Turbonomic instance, as follows:

- [Script Interface](#)

You installed Turbonomic version 8.1.5 or later as a new VM image (OVA or VHD; see [Installing on a Virtual Machine Image](#)). In this case, you can run the `enable_reporting.py` script to set up embedded reporting.

- [Editing the Turbonomic cr.yaml File](#)

You installed Turbonomic version 7.22.5 or later (up to version 8.1.5) as a new VM image.

OR

You installed Turbonomic as a Kubernetes cluster, not using the VM image for your Turbonomic installation.

In these cases, you manually edit the `charts_v1alpha1_x1_cr.yaml` file for your installation of Turbonomic.

### Note:

If you installed Turbonomic version 8.1.4 or earlier as a VM image, contact your support representative for help with enabling Embedded Reports.

## Using the Script Interface

If you installed Turbonomic as a VM image for version 8.1.5 or later, the script to enable Embedded Reports is already installed on your installation at `/opt/local/bin/enable_reporting.py`.

To run this script:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: [your\_private\_password]

2. Go to the script directory.

```
cd /opt/local/bin
```

3. Run the script.

```
./enable_reporting.py
```

The script prompts you for two passwords:

- Grafana admin password

This password enables access to Grafana from external URLs and also from the extractor component that feeds data to Grafana.

Do not use special characters.

**Important:**

This step is the only time that you should change the Grafana Admin password. If you later change the Grafana Admin password, the Embedded Reporting components cannot communicate properly with the other components in the platform, and you should contact your Support representative.

- Grafana database password

This password enables communication between Grafana and the Postgres database that stores the reporting data.

After you supply the passwords, the script displays a confirmation message similar to:

```
Successfully applied new changes to /opt/turbonomic/kubernetes/operator/deploy/crds/
charts_v1alpha1_xl_cr.yaml.
Backup written to /opt/turbonomic/kubernetes/operator/deploy/crds/
charts_v1alpha1_xl_cr.yaml.bak
```

This indicates that the script successfully updated the Turbonomic configuration. The script then applies the changed configuration to enable the Embedded Reports feature. It should display messages similar to:

```
Applying CR file /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
Warning: kubectl apply should be used on resource created by either
kubectl create --save-config or kubectl apply
xl.charts.helm.k8s.io/xl-release configured
Waiting for changes to take effect...
Restarting api pod to apply configuration changes.
pod "api-65cf47986f-jxszd" deleted
Changes have been successfully applied. Embedded reporting is now enabled.
```

4. Verify your installation.

Run the command:

```
./enable_reporting.py --validate
```

If Embedded Reports are successfully enabled, the script output should be:

```
No obvious embedded reporting installation errors detected.
```

## Editing the Turbonomic YAML File

These instructions describe how to locate and edit the `charts_v1alpha1_x1_cr.yaml` for the VM image installation. If you installed on a Kubernetes node cluster, then the file can be in a different location.

To enable Embedded Reports, complete the following tasks:

- Enable the processes that implement the embedded reporting.
- Update the API pod to enable new search and data ingestion capabilities.
- Double-check the installation.
- Optionally enable email subscriptions.

Enable the Grafana Exporter, TimescaleDB, and data extraction processes by editing the `charts_v1alpha1_x1_cr.yaml` file.

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: `turbo`
- Password: `[your_private_password]`

2. Open the following file in a text editor:

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

3. Specify the IP address of the Turbonomic instance for external access to the TimescaleDB database.

In the `global:` section of the file, add the following line, where **<Platform\_IP>** is the IP address of your instance:

```
global:
 externalTimescaleDBIP: <Platform_IP>
```

4. Enable the Grafana process.

Find the `grafana:` section in the `crds/charts_v1alpha1_x1_cr.yaml` file, and uncomment the line, `enabled: true`.

5. Enable Postgres as the database type.

Enabling Postgres sets persistent storage of historical data for Embedded Reports. In the `grafana:` section, find the subsection for `grafana.ini: database:` and uncomment the line, `type: postgres`.

The changes that you have made so far should be similar to:

```
global:
 externalTimescaleDBIP: <Platform_IP>
 ...

grafana:
 enabled: true
 adminPassword: admin
 grafana.ini:
 database:
 type: postgres
 ...
```

6. Change the admin and database passwords.

It is good practice to change any passwords, and not keep their default values.

### Important:

Use only alpha-numeric characters for these passwords.

These passwords enable communication between the various Embedded Reports components. Some of the components accept only alpha-numeric characters. If you use special characters, then the components cannot communicate. Further, the steps to correct these passwords require assistance from your Support representative.

To set the passwords:

- Set the Grafana admin password.

This password enables access to Grafana from external URLs and also from the extractor component that feeds data to Grafana. In the `grafana:` section, change the value of `adminPassword`.

Do not use special characters.

Assume that your password is `MyNewGrafanaPassword`. Then, you would set `adminPassword`:  
**MyNewGrafanaPassword**

**Important:**

This step is the only time that you should change the Grafana Admin password. If you later change the Grafana Admin password, the Embedded Reporting components cannot communicate properly with the other components in the platform, and you should contact your Support representative.

- Set the Grafana database password.

This password enables communication between Grafana and the Postgres database that stores the reporting data. In the `grafana:` section, find the subsection for `grafana.ini: database: password:` and change the password value.

7. Enable the reporting processes.

Just after the `properties:` section that you added, and at the same level to it, add the following entries to enable the reporting processes:

```
reporting:
 enabled: true
timescaledb:
 enabled: true
extractor:
 enabled: true
```

It is important that you align these entries with the indentation for the `grafana:` section and the `properties:` section. The changes that you have made should now be similar to:

```
global:
 externalTimescaleDBIP: <Platform_IP>
 ...

grafana:
 enabled: true
 adminPassword: MyNewGrafanaPassword
 grafana.ini:
 database:
 type: postgres
 password: MyNewDatabasePassword

properties:
 extractor:
 grafanaAdminPassword: MyNewGrafanaPassword

reporting:
 enabled: true
timescaledb:
 enabled: true
extractor:
 enabled: true
```

8. When you are done editing the `charts_v1alpha1_x1_cr.yaml` file, save and apply your changes.

- Save your changes and quit the text editor.

- Run the command to apply the changes.

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

- Delete the api and extractor pods.

Deleting these pods triggers them to restart, which loads the changes you made.

Run the following command to get the full pod names, run the command,

```
kubectl get pods -n turbonomic
```

Find the entries for the pods that begin with api and extractor. For example, assume that the entries are:

```
...
api-7887c66f4b-shndq 1/1 Running 0
...
extractor-5b86976bc8-vxwz4 1/1 Running 0
...
```

Run the following commands:

```
kubectl delete pod -n turbonomic api-7887c66f4b-shndq
```

```
kubectl delete pod -n turbonomic extractor-5b86976bc8-vxwz4
```

9. In an SSH terminal session for your Turbonomic instance, set the password for Postgres and enable access to psql on the appliance's localhost. Run the following commands:

```
sudo -iu postgres psql -c "ALTER ROLE postgres WITH PASSWORD 'vmturbo'"
```

```
sudo -iu postgres psql -c "CREATE ROLE turbo WITH SUPERUSER CREATEDB CREATEROLE LOGIN
REPLICATION BYPASSRLS PASSWORD 'vmturbo'"
```

10. Verify your installation.

- Verify that the Embedded Reports pods are running.

Run the following command:

```
kubectl get pods -n turbonomic
```

The output should include entries similar to:

| NAME                       | READY | STATUS  | RESTARTS |
|----------------------------|-------|---------|----------|
| extractor-7759dbcb47-vs6hr | 1/1   | Running | 0        |
| grafana-84ccb4bfb-17sp7    | 1/1   | Running | 0        |

- Verify that Postgres is running.

The Postgres database should be running as a daemon on the Turbonomic server machine. To check the status, run the command:

```
sudo systemctl status postgresql-12.service
```

You should see output similar to:

```
postgresql-12.service - PostgreSQL 12 database server
Loaded: loaded (/usr/lib/systemd/system/postgresql-12.service; enabled; vendor preset:
disabled)
Active: active (running) since Wed 2020-07-29 06:39:43 UTC; 14h ago
Docs: https://www.postgresql.org/docs/12/static/
Process: 1536 ExecStartPre=/usr/pgsql-12/bin/postgresql-12-check-db-dir ${PGDATA}
(code=exited, status=0/SUCCESS)
Main PID: 1562 (postmaster)
Tasks: 15
```

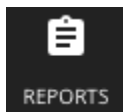
```

Memory: 145.5M
CGroup: /system.slice/postgresql-12.service
├─ 419 postgres: TimescaleDB Background Worker Scheduler
├─ 1562 /usr/pgsql-12/bin/postmaster -D /var/lib/pgsql/12/data/
├─ 1928 postgres: logger
├─ 1986 postgres: checkpointer
├─ 1988 postgres: background writer
├─ 1989 postgres: walwriter
├─ 1990 postgres: autovacuum launcher
├─ 1991 postgres: stats collector
├─ 1992 postgres: TimescaleDB Background Worker Launcher
├─ 1994 postgres: logical replication launcher
├─ 4054 postgres: grafana_backend grafana 10.233.90.172(33038) idle
├─ 4884 postgres: grafana_backend grafana 10.233.90.172(35814) idle
├─ 4912 postgres: grafana_reader extractor 10.233.90.172(33898) idle
├─ 11365 postgres: grafana_reader extractor 10.233.90.172(40728) idle
└─ 32367 postgres: TimescaleDB Background Worker Scheduler

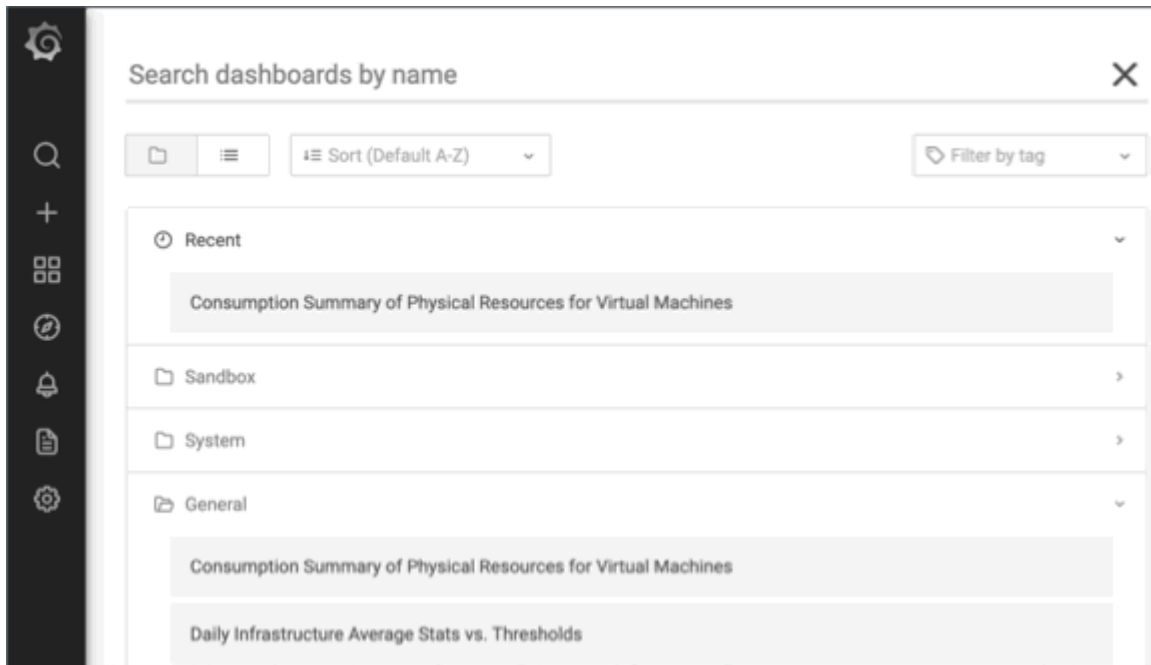
```

## Navigating to the Embedded Reports Page

After you have completed the steps to enable Embedded Reports, open the Turbonomic user interface and click **Reports** in navigation bar.



This opens dashboards and charts in a new browser tab.



You can search for specific dashboards or browse folders to find the dashboards you want. You can also create custom reports.

Dashboards and charts are powered by the Grafana® observability platform. With Grafana, it's easy to navigate the existing dashboards, and to make your own charts and dashboards with no coding required. If you are new to Grafana and need help getting started, read the documentation available at:

<https://grafana.com/docs/grafana/latest/>

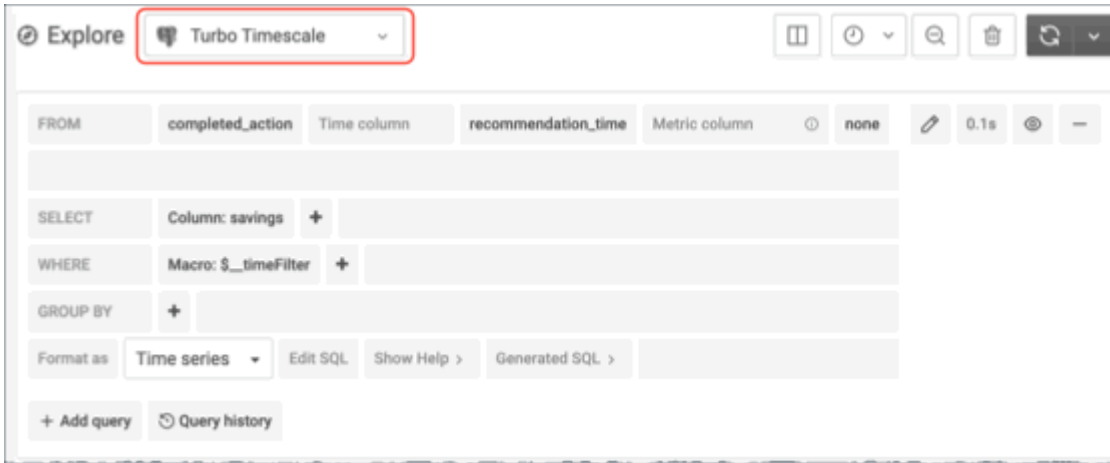
### Note:

Starting with version 8.6.1, Turbonomic no longer has an agreement to use the Enterprise license of Grafana, and ships the open-source Community license instead. After you update an earlier version of Turbonomic to version 8.6.1 or later, Grafana automatically switches to the Community license and you

should be able to use Embedded Reporting as usual. However, be aware that the Community license does not support PDF reports. Please contact your Turbonomic representative for guidance on PDF reports.

## Creating Custom Reports

To create custom reports, you must run SQL Server queries against the Embedded Reports database (Turbo Timescale).



The database schema includes certain tables against which you can run queries. For detailed documentation for the schema, see [Integration - Embedded Reporting](#).

## (Optional) Report Editing

To create and edit reports, a user must have Report Editor privileges. This user can have any role, as long as it is not a *shared* or *scoped* role. Only one user per Turbonomic instance is allowed to have Report Editor privileges (by default, the local **administrator** user). You can grant these privileges to another user.

To create a user account with Report Editor privileges:

1. In the Turbonomic user interface, navigate to **Settings / User Management**.
2. Choose the user account that you want to configure as a Report Editor. You can either edit an existing account or create a new one.
3. Choose a role for the user. The user can have any role, as long as it is not a *shared* or *scoped* role.
4. Under **Options**, choose **DESIGNATE AS REPORT EDITOR**.
5. Set any other properties for the user account that you want, and save the user account.

### Note:

It can take up to 30 minutes before the Reports page shows the Report Editor username. This usually occurs if you have changed the Report Editor multiple times.

To access reports from the user interface, a user must have the Administrator or Site Administrator role, or a non-administrator role without a defined scope. For example, a user with the Observer role but without a scope can access reports.

The default Shared Observer and Shared Adviser roles require scopes, so users with these roles cannot access reports.

## Embedded Reports Storage Requirement Estimates

The Embedded Reports feature uses a TimescaleDB server to manage the chart data. This is a PostgreSQL server running with the TimescaleDB extension. You must configure the datastore for your Turbonomic instance so it has enough space to support the TimescaleDB requirements.



When you initially enable Embedded Reports, you should estimate the storage you will need, and configure the platform storage accordingly. If you have already enabled Embedded Reports, you should check your current storage configuration and decide whether it meets your needs now and into the future.

The storage that your TimescaleDB requires depends on:

- Data retention period

How long to store the TimescaleDB data.

- The size of your environment

The count of entities Turbonomic manages in your environment. This count changes over time. You should think of it as the average number of entities in your environment over the given data retention period.

Also note that increased entity count increases the data requirement, as does other activity. Storage requirements can increase over time for reasons such as:

- You add entities such as workloads, application components, storage, or hosts to your environment.
- You configure new targets.

## Storage Estimates Lookup Tables

We have investigated the TimescaleDB storage requirements for different topologies and retention periods. The following table lists the estimates that we have calculated. Please be aware that your environment could have different requirements.

|                  | Number of Entities |       |       |       |       |       |       |
|------------------|--------------------|-------|-------|-------|-------|-------|-------|
| Retention Period | 10k                | 25k   | 50k   | 100k  | 250k  | 500k  | 1000k |
| 6 months         | 36GB               | 91GB  | 182GB | 364GB | 910GB | 1.8TB | 3.6TB |
| 1 year           | 72GB               | 181GB | 361GB | 723GB | 1.8TB | 3.5TB | 7TB   |
| 2 years          | 144GB              | 361GB | 721GB | 1.4TB | 3.5TB | 7.2TB | 14TB  |

Note that the default installation grants a disk quota of 200GB to the TimescaleDB. For the default installation, we estimate that the database can support the following entity counts:

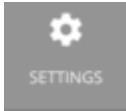
| Retention Period | Entity Count |
|------------------|--------------|
| 6 months         | 55k          |
| 1 year           | 27k          |
| 2 years          | 14k          |

## Setting the Data Retention Period

By default, Turbonomic sets the retention period for Embedded Reports to be 365 days. You can see the currently set retention period, and change it in the Turbonomic user interface.

To execute these actions, navigate to the Maintenance Options page:

1. Navigate to the Settings Page.



Click to navigate to the Settings Page.

2. Choose Maintenance Options.



#### Maintenance Options

3. Set the data retention period for Embedded Reports.

In the **Data Retention** group of controls, find the field for **Saved Reporting Data**. This displays the current data retention period for Embedded Reports, in days. The default is 365 days.

To change the retention period, enter a different number of days, and then click **Apply Settings**.

## Increasing Storage Capacity for TimescaleDB

If you estimate the storage requirements for Embedded Reports *after* you have installed Turbonomic, you might learn that you need to increase the storage capacity that is available to the TimescaleDB.

The platform uses Logical Volume Management (LVM) to manage the VM disks. To increase database storage, you should add a new disk to the VM, and then use it to extend the LVM logical volume, `/dev/turbo/var_lib_mysql`. This logical volume serves both the historical database and the Embedded Reports database. When you have done that, you will increase the quota for the TimescaleDB.

For more information, see [Increasing Available Disk Space](#)

## Estimating Entity Count

To get a sense of entity count in your environment, search the Turbonomic log file for an INFO message that lists the number of entities for each discovery cycle. You can search for the string `INFO [Stages$BroadcastStage]`. The INFO string should be similar to:

```
topology-processor-6f6486df64-zf 2021-09-27 20:51:33,724 INFO
[Stages$BroadcastStage] : Successfully sent 1505 entities within topology...
```

This example shows that you have 1505 entities in the topology. You should consider how your inventory changes over time. For example, you can check the entity count over time to see whether it increases regularly.

## (Optional) Enabling the Data Exporter

To support Data Export, Turbonomic provides an extractor component that can stream data to a standard format. You can load that data into search and analytics services such as Elasticsearch.

To enable the Data Exporter, you must:

- Enable the extractor component.

The extractor is a component that runs as part your Turbonomic installation. The extractor is not enabled by default.

- Deploy a connector that delivers the extractor's stream to your data service.

The extractor publishes Turbonomic data as Kafka topics. The connector enables your data service to consume the data topic. This document includes a deployment file for a sample Elasticsearch connector.

## Enabling the Extractor Component

The first step to enabling the Data Exporter is to enable the extractor component. To enable the extractor:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Username: [your\_private\_password]

2. Edit the cr.yaml file to enable the extractor component.

In the same SSH session, open the cr.yaml file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

3. Edit the entry for the extractor component.

### Note:

If you have enabled Embedded Reporting, then the extractor component will already be enabled (set to `true`).

You should understand that it is possible to enable the Data Exporter without enabling Embedded Reports, just as it is possible to enable Embedded Reports without enabling the Data Exporter.

Search for the extractor entry in the cr.yaml file. It should appear as:

```
extractor:
 enabled: false
```

Change the entry to `true`.

4. Edit the entry for the extractor properties.

Search for the extractor entry in the cr.yaml file. It should appear as:

```
properties:
 extractor:
 enableDataExtraction: false
```

Change the entry to `true`.

5. Save and apply your changes to the platform.

After you save your changes, use `kubectl` to apply the changes:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

6. Verify that the extractor component is running.

Give the platform enough time to restart the components. Then execute the command:

```
kubectl get pods -n turbonomic
```

You should see output similar to the following:

| NAME                       | READY | STATUS  | RESTARTS |
|----------------------------|-------|---------|----------|
| ...                        |       |         |          |
| extractor-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                        |       |         |          |

Look for an entry for the extractor component. If the entry is present, then the extractor component is installed and running.

## Deploying a Connector

The extractor publishes Turbonomic data as Kafka topics. To load this data into a search and analysis service, you must deploy a connector to that service. For example, if you want to load the data into Elasticsearch, then you must deploy an Elasticsearch connector.

You deploy the connector in the same Kubernetes node that runs the Turbonomic platform. To do this, create a Kubernetes *Deployment* that declares the pods you need for the connector. Below, you can see a sample deployment of a connector to Elasticsearch.

To deploy the connector, you create a deployment yaml file on the same host that is running the extractor component, and execute the command:

```
kubectl create -f <MyConnectorDeployment.yaml>
```

Where **<MyConnectorDeployment.yaml>** is the name of the deployment file.

Assume the name of the deployed pod is `es-kafka-connect`. To verify that the connector is running, execute `kubectl get pods -n turbonomic`. You should see output similar to:

| NAME                              | READY | STATUS  | RESTARTS |
|-----------------------------------|-------|---------|----------|
| ...                               |       |         |          |
| es-kafka-connect-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                               |       |         |          |

After you deploy the connector, wait for a cycle of Turbonomic analysis (approximately ten minutes). Then you should be able to see the entities and actions from your Turbonomic environment, loaded as JSON in your data service.

## Connector Deployment Sample

Assume that you want to deploy a connector to Elasticsearch so that service can process the exported data. For example, you could use Kibana with Elasticsearch to display data dashboards. Let's say you have:

- Deployed Elasticsearch to a VM on the network where you are running Turbonomic. The Elasticsearch host is visible from the Turbonomic Kubernetes node. You will specify this host address in the connector deployment.
- Set up an Elasticsearch index to load the Turbonomic data. You will specify this index in the connector deployment.

The following listing is a deployment that uses a Logstash image to collect the extractor data and pipe it to the Elasticsearch host. The deployment also sets up storage volumes, configures the input from the extractor, and configures output to the Elasticsearch instance.

As you go over the listing, pay attention to the following:

- The location of the Elasticsearch host and the login credentials:

```
...
 env:
 - name: ES_HOSTS
 value: "<UrlToMyElasticsearchHost>"
 - name: ES_USER
 value: "<MyElasticsearchUser>"
 - name: ES_PASSWORD
 valueFrom:
 secretKeyRef:
 name: <MyES_KeyName>
 key: <MyES_Key>
 ...
```

Logstash will use the following environment variables:

- `ES_HOSTS`: to identify where to pipe the exported data.
- `ES_USER`: to identify the user account on Elasticsearch.

- ES\_PASSWORD: for the account login. This connector example assumes that you have stored the Elasticsearch password as a Kubernetes Secret.

Logstash uses the ES\_HOSTS environment variable to identify where to pipe the exported data.

- The name of the Kafka topic:

```
...
 logstash.conf: |
 input {
 kafka {
 topics => ["turbonomic.exporter"]
 }
 }
 ...
```

The Logstash input configuration expects a single topic named `turbonomic.exporter`.

- The Logstash output configuration is to the Elasticsearch server that is identified by the ES\_HOSTS environment variable. You specify your own Elasticsearch index in place of **<MyElasticsearchIndex>**

```
...
 output {
 elasticsearch {
 index => "<MyElasticsearchIndex>"
 hosts => ["${ES_HOSTS}"]
 }
 }
 ...
```

### Sample Listing: Elasticsearch Connector

This listing is a sample of a deployment file that can work to create an Elasticsearch connector for the Data Exporter. Note that you will need to change some settings, such as username and password. You also might need to specify ports and other settings to make the connector comply with your specific environment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: elasticsearch-kafka-connect
 labels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
spec:
 replicas: 1
 selector:
 matchLabels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
 template:
 metadata:
 labels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
 spec:
 containers:
 - name: logstash
 image: docker.elastic.co/logstash/logstash:7.10.1
 ports:
 - containerPort: 25826
 env:
 - name: ES_HOSTS
 value: "<UrlToMyElasticsearchHost>"
 - name: ES_USER
 value: "<MyElasticsearchUser>"
 - name: ES_PASSWORD
 valueFrom:
 secretKeyRef:
 name: <MyES_KeyName>
 key: <MyES_Key>
 resources:
 limits:
 memory: 4Gi
 volumeMounts:
 - name: config-volume
 mountPath: /usr/share/logstash/config
 - name: logstash-pipeline-volume
 mountPath: /usr/share/logstash/pipeline
 volumes:
```

```

- name: config-volume
 configMap:
 name: logstash-configmap
 items:
 - key: logstash.yml
 path: logstash.yml
- name: logstash-pipeline-volume
 configMap:
 name: logstash-configmap
 items:
 - key: logstash.conf
 path: logstash.conf

apiVersion: v1
kind: ConfigMap
metadata:
 name: logstash-configmap
data:
 logstash.yml: |
 http.host: "0.0.0.0"
 path.config: /usr/share/logstash/pipeline
 logstash.conf: |
 input {
 kafka {
 topics => ["turbonomic.exporter"]
 bootstrap_servers => "kafka:9092"
 client_id => "logstash"
 group_id => "logstash"
 codec => "json"
 type => "json"
 session_timeout_ms => "60000" # Rebalancing if consumer is found dead
 request_timeout_ms => "70000" # Resend request after 70 seconds
 }
 }
 filter {
 }
 output {
 elasticsearch {
 index => "<MyElasticsearchIndex>"
 hosts => ["${ES_HOSTS}"]
 user => "${ES_USER}"
 password => "${ES_PASSWORD}"
 }
 }
}

apiVersion: v1
kind: Service
metadata:
 labels:
 app: elasticsearch-kafka-connect
 name: elasticsearch-kafka-connect
spec:
 ports:
 - name: "25826"
 port: 25826
 targetPort: 25826
 selector:
 app: elasticsearch-kafka-connect

```

## (Optional) Changing the IP Address of the Platform Node

For standard installations of Turbonomic (installed as a VM image), you might need to change the platform's IP address. For example, if you have to move the VM then you might need to assign it a different address. If you must change the IP address of the platform, you can use the supplied scripts.

### Note:

You should change the IP address of your Turbonomic installation as seldom as possible. This is a sensitive action that can impact unforeseen dependencies.

You can safely use the following steps to change your IP address if your installation satisfies one of the following conditions:

- You initially installed the platform OVA as version 8.0.6 or later.
- You have updated your Turbonomic to version 8.0.7 or later.

If your installation does not meet either one of these conditions, *do not use the following instructions to change the IP address*. If you must change your IP address and you cannot update to version 8.0.7 or later, contact your support representative.

To change the IP address of the Turbonomic VM:

1. Get your information ready.

Identify both the current IP address for your platform, and the new IP address you will use.

You must also know the credentials to open a shell session on the VM and run commands.

2. Create a full snapshot of the VM.

It is important to make a full snapshot of your installation before you try to modify its IP address.

3. Change the VM's IP address.

The Turbonomic VM includes the `ipsetup` script to perform this task.

- a. Open an SSH terminal session to your Turbonomic VM.

Use the following credentials:

- Username: `turbo`
- Password: Give the password that you assigned the `turbo` account when you first installed the platform.

- b. Once the session is open, execute the `ipsetup` script:

```
sudo /opt/local/bin/ipsetup
```

When the script runs it requests the following inputs.

**Note:**

*You must provide values for these required fields.* Otherwise the installation can fail or your VM can be unreachable:

- **Required:** Do you want to use DHCP or set a static IP...  
Choose `static`
- **Required:** Please enter the IP Address for this machine
- **Required:** Please enter the network mask for this machine
- **Required:** Please enter the Gateway address for this machine
- **Required:** Enter DNS Server(s) IP Address for this machine

You should make a note of the IP address that you provide.

- c. Propagate your IP change through to the Kubernetes cluster on the VM.

```
sudo /opt/local/bin/kubeNodeIPChange.sh
```

- d. Verify that the change is successful.

Log into the Turbonomic user interface for the newly located installation, and ensure that it displays correctly. You should review the Supply Chain, your groups, and your policies. You should also ensure that charts show data correctly.

When you are sure that the change is successful, you can remove the snapshot you made of the VM in its old location.

## (Optional) Enabling and Disabling Probe Components

In Turbonomic, a probe is a platform component that connects to a target. It discovers the target's entities and loads them into the Turbonomic supply chain, and it can execute actions on the devices in

the target environment. Turbonomic ships with a large number of probe components that you can use to connect Turbonomic with your environment.

When you first install Turbonomic, it enables a certain set of probes by default, and leaves other disabled. Each probe consumes resources in your Turbonomic installation. If there are any probes that you do not need, then you should consider disabling them. On the other hand, if there are disabled probes that you do need, you must enable them to put them into service.

**Note:**

As Turbonomic evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

## Viewing the Current List of Available Probes

As you update your version of Turbonomic, more probes can come available with the update. However, the update does *not* modify your current configuration of enabled or disabled probes. This means that any new probes that come with an update will not be available to you by default.

To enable any new probes, you must first know the internal name for the probe. To get a list of probes that are *available* to your current version, you can view the contents of the `values.yaml` file.

1. Open an SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: `turbo`
- Username: `[your_private_password]`

2. Display the list of available probes.

```
cat /opt/turbonomic/kubernetes/operator/helm-charts/x1/values.yaml
```

The results should be similar to:

```
customdata:
 enabled: false
dynatrace:
 enabled: false
gcp:
 enabled: false
hpe3par:
 enabled: false
...
```

This list gives the internal names of the probes. If you want to add a new probe to your list of configured probes, you must use the internal name, and set `enabled: true`.

## Viewing the Current List of Configured Probes

Your current installation of Turbonomic has a certain set of available probes. Some of these will be enabled, and it is likely that some probes are disabled. To View the current configuration of probes that are currently available, open the `cr.yaml` file for your Turbonomic installation and review the probe entries:

1. In the same SSH session, open the `cr.yaml` file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

2. Search for the list of probes



This will include all the probes that are configured for your current installation. The list will be similar to:

```
actionscript:
 enabled: true
appdynamics:
 enabled: true
appinsights:
 enabled: true
aws:
 enabled: true
azure:
 enabled: true
dynatrace:
 enabled: true
hpe3par:
 enabled: true
horizon:
 enabled: false
hyperflex:
 enabled: false
...
```

This list identifies all the probes that are currently configured for your installation, and shows whether they are enabled (`true`) or disabled (`false`).

#### Note:

This list of probes is not identical to the list of probe *Pods* that are running in your installation. Some probes use multiple pods. Probe pod names use the following convention, where **{ProbeName}** is the probe internal name (in the lists above), and **{NameExtension}** is an optional extension to that name in case there are multiple pods for this probe:

```
mediation-{ProbeName}{NameExtension}
```

For example, if you execute `kubectl get pods -n turbonomic`, the results can show the following for the `vcenter` probe:

| NAME                                       | READY | STATUS  | RESTARTS |
|--------------------------------------------|-------|---------|----------|
| mediation-vcenter-5bc4f5fbd4-nzm4j         | 1/1   | Running | 0        |
| mediation-vcenterbrowsing-5c5987f66c-bfjq4 | 1/1   | Running | 0        |

## Enabling/Disabling Probes

To enable or disable probes in Turbonomic, you will edit the `cr.yaml` file to add new probes and to change the values of the `enabled:` properties. Then you will apply those changes to reload the platform components.

1. Follow the steps above to log into a SSH terminal session for your Turbonomic instance and display the list of available probes.
2. In the SSH session, open the `cr.yaml` file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

3. Edit the probe entries.

To enable or disable currently configured probes, find the probes you want to edit and change the settings to enable or disable them.

To add new probes to the list, copy the probe entry you want from the output when you used `cat` to view the available probes. Then paste that entry into the `cr.yaml` file and set `enabled: true`.

4. Save and apply your changes to the platform.

#### Note:

During the online or offline upgrade process, you should not use `kubectl` to apply these changes at this time.

After you save your changes, use `kubectl` to apply the changes:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

5. Verify that the probes have installed correctly and all the Turbonomic pods have started.

Execute `kubectl get pods -n turbonomic` and review the list for the mediation pods that implement your probes. Note that all pods should display `READY` and `STATUS` states similar to:

| NAME  | READY | STATUS  | RESTARTS |
|-------|-------|---------|----------|
| [...] | 1/1   | Running | 0        |

6. View the new probe configuration in the user interface.

Refresh your browser and navigate to the Target Management page. You should now see the target categories and types to match your configuration changes.

## (Optional) Disabling Component Health Monitoring

The Notification Center includes component health information to notify you of Turbonomic component issues and the steps you can take to resolve these issues. To open Notification Center, click the information icon on the main menu.

If you would rather use your own tools to monitor the Turbonomic component status, you can disable the feature in the CRD file.

To disable component health monitoring:

1. Open an SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: `turbo`
- Username: `[your_private_password]`

2. Open the following `cr.yaml` file in a text editor:

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

3. Add the following to the CRD file:

```
properties:
 global:
 featureFlags:
 componentHealth: false
```

4. When you are done editing the `cr.yaml` file, save and apply your changes.

- Save your changes and quit the text editor.
- To apply the changes, run the following command in your SSH terminal session:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

5. Delete the api and Cluster Manager pods.

Deleting these pods triggers them to restart, which loads the changes you made.

- Close the Turbonomic UI if it is open in your browser.
- To get the full pod names, run the command:

```
kubectl get pods
```

- Find the entries for the pods that begin with `api` and `clustermgr`. For example, assume that the entries are:

```
...
 api-7df7b8c8c7-6rqgg
...
 clustermgr-7c6579745d-m8dwc
...
```

- Run the following command to delete the pods that use the `api` and `clustermgr` pod names:

```
kubectl delete pods api-7df7b8c8c7-6rqgg clustermgr-7c6579745d-m8dwc
```

**Note:**

The pod names change when deleted and started again so always get the latest pods before you run the delete command.

The pods are deleted and re-created.

6. Verify the deletion and recreation of the pods.

To verify that the pods are re-created and running, run the command:

```
kubectl get pods
```

After all of the pods start, the `READY` column should read `1/1`, `2/2`, and so on and the `STATUS` column should read `Running` for each pod.

The output should include entries similar to this example:

```
clustermgr-7c6579745d-m8dwc 1/1 Running 1 (2d16h ago) 3d15h
api-7df7b8c8c7-6rqgg 1/1 Running 1 (2d16h ago) 3d15h
```

## (Optional) Overriding Default Ports

MariaDB or MySQL database server uses port 3306 by default. Postgres database (for Embedded Reporting) uses port 5432 by default. Use the following sample configurations to override the default ports.

Sample configuration for MariaDB with IP address used for the database server with custom port (non-3306):

```
global:
 repository: icr.io/copen/turbonomic
 tag: 8.9.1
 externalIP: 10.10.10.10
 externalDbIP: 10.10.10.11
 externalDbPort: 3307
```

Sample configuration for MariaDB with a fully qualified domain name used for the database server with custom port (non-3306):

```
global:
 repository: icr.io/copen/turbonomic
 tag: 8.9.1
 externalIP: 10.10.10.10
 externalDBName: MyDatabaseServer.com
 externalDbPort: 3307
```

Sample configuration for Postgres database (for Embedded Reporting) with custom port (non-5432):

```
global:
 repository: icr.io/copen/turbonomic
 tag: 8.9.1
 externalDbIP: 10.10.10.30
```

externalTimescaleDBIP: 10.10.10.31  
externalTimescaleDBPort: 5434

## License Installation and First-time Login

Before you begin, make sure you have your full or trial license key file that was sent to you in a separate email. Save the license file on your local machine so you can upload it to your Turbonomic installation.

To use Turbonomic for the first time, perform the following steps:

1. Type the IP address of your installed Turbonomic instance in a Web browser to connect to it.
2. Log in to Turbonomic.
  - Use the default credential for **USERNAME**: administrator.
  - Type a password for **PASSWORD**.
  - Type the password again to verify it for **REPEAT PASSWORD**.
  - Click **CONFIGURE**.
3. Continue setting up your Turbonomic installation.

Click **LET'S GO**.

4. Open the **Enter License** fly-out.

Click **IMPORT LICENSE**.

5. Upload your license key file.

a. In the Enter License fly-out, you can upload the license in one of the following ways:

- Drag and drop the license key file into the Enter License fly-out.
- Browse to the license key file.

Be sure to upload only .xml, .lic, or .jwt files.

b. Click **SAVE**.

Depending on which license you have installed, the license enables either a trial or a full unlimited license for Turbonomic.

## Tracking License Usage with IBM License Service

IBM® License Service collects and measures Turbonomic license usage. Use the License Service to track your license usage and periodically generate audit snapshots to fulfill licensing requirements.

License Service is installed in OVA automatically when you run the `t8cInstall.sh` script. No manual processes are required. License service installs in its own namespace, `ibm-common-services`, rather than in the Turbonomic namespace. When you run `kubectl get pods -n ibm-common-services`, you can see the two pods that are running in this namespace: `ibm-licensing-operator` and `ibm-licensing-service-instance`, as shown in the following example:

| NAME                                                         | READY | STATUS  | RESTARTS | AGE   |
|--------------------------------------------------------------|-------|---------|----------|-------|
| <code>ibm-licensing-operator-5fb7fcfbfc-wpc9q</code>         | 1/1   | Running | 0        | 2d21h |
| <code>ibm-licensing-service-instance-5945cd746b-ds9cg</code> | 1/1   | Running | 0        | 2d21h |

### How to use License Service

After you obtain the required API token, you can use the License Service user interface (UI) in your browser to retrieve license usage data for monitoring and compliance. You can also retrieve an audit snapshot of the data to use as audit evidence.

- Obtaining an API token

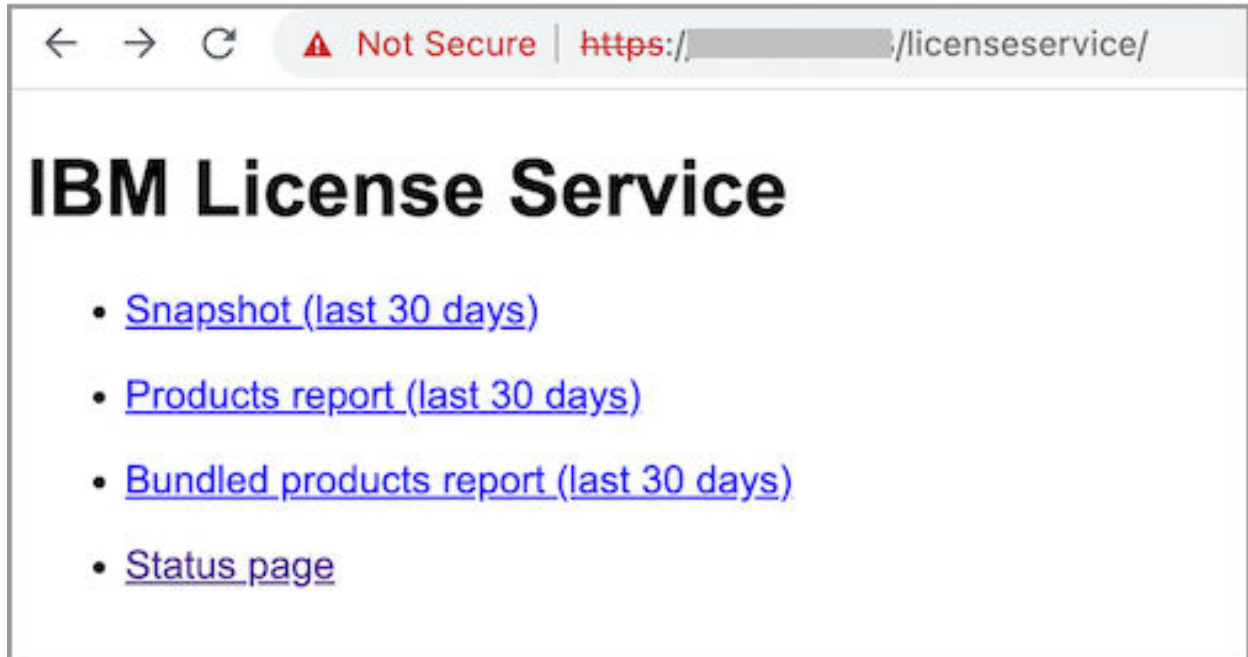
First, use the command line to obtain the License Service API token to authenticate and retrieve license usage data. The following code shows an example Linux command and the API token that is returned.

```
[turbo@node1 ~]$ kubectl get secret ibm-licensing-token -o jsonpath={.data.token} -n ibm-common-services | base64 -d
<token_value>
```

For more information, see [Obtaining and updating an API token](#).

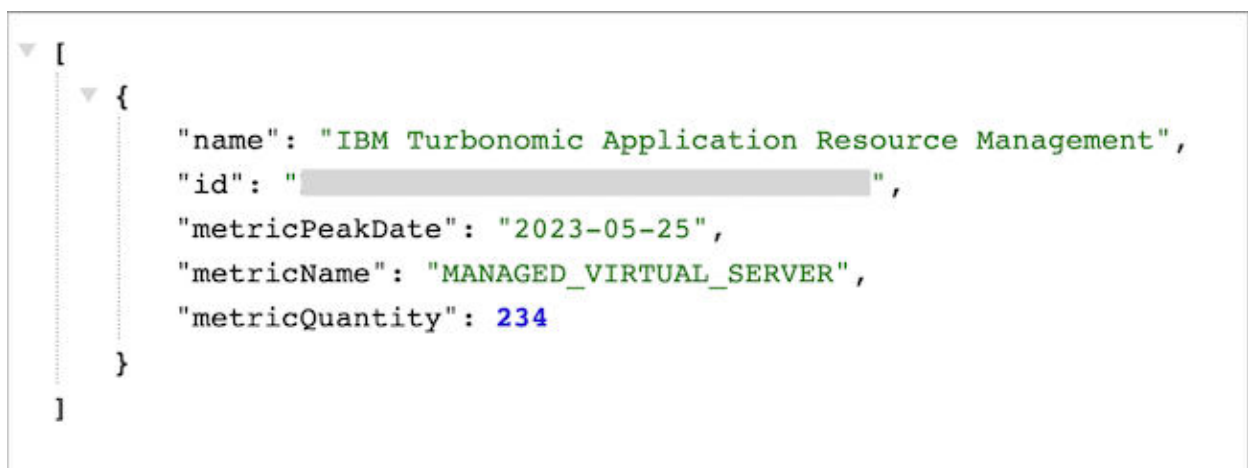
- Accessing the License Service UI

In OVA, you can access the License Service UI by going to the `https://<turbonomic_ip_address>/licenseservice` URL in your browser, as shown in the following screen capture.



- Retrieving license usage data

In the License Service UI, click **Products report (last 30 days)** to retrieve Turbonomic license usage data. You can also provide optional parameters in the URL, such as `startDate` and `endDate`. The following screen capture shows a sample response from the License Service UI.



For more information about each field in the response, see [Retrieving license usage of products](#).

- Generating an audit snapshot

Audit snapshots are needed for compliance and audit purposes. You do not need to complete any manual actions to prepare the audit snapshot, you need only to generate it.

Click **Snapshot (last 30 days)** in the License Service UI and enter your API token to generate and download a .zip file that contains the audit snapshot.

You can also generate the snapshot from command line by running a `curl` command that is similar to the following example.

```
curl -k -X GET
 https://<turbonomic_ip_address>/licenseservice/snapshot?
token=<token_value>&startDate=2023-05-01&endDate=2023-05-30
 --output audit_snapshot.zip
```

Alternatively, you can type the audit snapshot URL directly in your browser, substituting your Turbonomic IP address and API token values, as shown in the following example.

```
https://<turbonomic_ip_address>/licenseservice/snapshot?
token=<token_value>&startDate=2023-05-01&endDate=2023-05-30
```

If you do not specify the `startDate` or `endDate` parameter, the default snapshot duration is the last 30 days.

For more information, see [Retrieving an audit snapshot](#).

## Single Sign-On Authentication

---

If your company policy supports Single Sign-On (SSO) authentication, you can configure Turbonomic to support SSO authentication via either Security Assertion Markup Language (SAML) 2.0 or OpenID Connect 1.0.

At a high-level, to do this you will:

- Create external groups or at least one external user for SSO. See [Managing User Accounts](#).
- Configure Turbonomic to use SSO authentication.

You will configure one of:

- SSO via a SAML Identity Provider (IdP). See [Setting Up SAML Authentication](#).
- SSO via an OpenID Identity Provider. See [Setting Up OpenID Authentication](#).

This section describes how to configure Turbonomic to use either SAML or OpenID to support SSO.

When SSO is enabled, users will provide their SSO credentials to log in to the Turbonomic instance. Once SSO is enabled, users cannot give local or Active Directory (AD) credentials for to login. The Identity Provider (IdP) will perform the authentication.

### Prerequisites

Before you begin, make sure the IdP is set up for SSO. You can use a proprietary or public IdP. For examples of settings for a public Okta IdP, see [What Are the Typical Settings for an IdP?](#).

## Setting Up SAML Authentication

Security Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication and authorization data between parties. To configure Turbonomic to authenticate via SAML:

1. (Required) Create external groups or at least one external user for SSO.

#### Important:

When SSO is enabled, Turbonomic only permits logins via the SSO IdP. Whenever you navigate to your Turbonomic installation, it redirects you to the SSO Identity Provider (IdP) for authentication before displaying the Turbonomic user interface.

Before you enable SSO for your Turbonomic installation, *you must configure at least one SSO user with Turbonomic administrator privileges*. If you do not, then once you enable SSO you will not be able to configure any SSO users in Turbonomic. To authorize an SSO user as an administrator, use **EXTERNAL AUTHENTICATION** to do one of the following:

- Configure a single SSO user with administrator authorization.  
Add an external user. The username must match an account that is managed by the IdP.
- Configure an SSO user group with administrator authorization.  
Add an external group. The group name must match a user group on the IdP, and that group must have at least one member.

For information about creating external groups or external users for SSO, see [Managing User Accounts](#).

2. (Required) Ensure that chrony is configured and the system time on your Turbonomic instance is correct.

For instructions, see [Synchronizing Time](#).

3. Obtain the metadata from your IdP.

You will use this metadata to configure SSO in the Turbonomic CR file located at:

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

To get the metadata:

- a. Contact your security administrator to obtain the metadata from IdP.
- b. Save the metadata file in a directory on your local machine. For example, save the file to:

```
/tmp/MySamlMetadata.txt
```

- c. Compare your metadata to the sample provided in [Example of IdP Metadata](#).

Cat out the file you just saved. It should be similar to the provided sample.

4. Obtain a certificate from IdP.

Contact your security administrator to obtain a certificate from IdP.

5. Update the CR file with your SAML configuration.

You now have the data that you need to configure SSO via SAML. You will edit the `cr.yaml` file that configures your Turbonomic node, and then deploy or restart the node.

- Display the contents of your downloaded SAML metadata.

For example, assuming you saved the file to this location on your local machine, execute the command:

```
cat /tmp/MySamlMetadata.txt
```

- Open the CR file for editing.

In a shell, `cd` to the `deploy/crds` directory in the Turbonomic VM:

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```

Then open the CR file for editing. For example, to open the file in VI:

```
vi charts_v1alpha1_x1_cr.yaml
```

As you edit this file, you will refer to the metadata that you obtained from your IdP.

- In the CR file, navigate to the entry for the API component.

In the CR file search for or scroll to the entry:

```
apiVersion: charts.helm.k8s.io/v1alpha1
```

You will make changes to this component spec, under `spec:properties:api:`

- Turn on the SAML feature.

For the first API property, set the following:

```
samlEnabled: true
```

- Set the SSO endpoint

In the SAML metadata, find the entry for `md:SingleSignOnService`. Within that element, find the `Location` attribute. The value of `Location` is the SSO endpoint. Using the sample metadata we have provided, you would make the following setting in your CR file:

```
samlWebSsoEndpoint: https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9MhzqiC30h7/sso/saml
```

- Set the SAML entity ID

In the SAML metadata, find the entry for `md:EntityDescriptor`. Within that element, find the `entityID` attribute. Using the sample metadata we have provided, you would make the following setting in your CR file:

```
samlEntityId: http://www.okta.com/exkexl6xc9MhzqiC30h7
```

- Set the SAML registration

Set the following property:

```
samlRegistrationId: simplesamlphp
```

- Set the SAML SP entity ID

Set the following property:

```
samlSpEntityId: turbo
```

- Enter the SAML certificate

In the metadata that you got from your IdP, find the entry for `<ds:X509Certificate>`. Copy the contents of this tag – copy the characters that are between `<ds:X509Certificate>` and `</ds:X509Certificate>`.

Create an entry for the certificate in the API properties section of the CR file. On a new line, enter:

```
samlIdpCertificate: |
```

Then open a new line after the entry you just created, and paste the certificate content that you copied from your metadata file.

The finished API section of the CR file should be similar to the following:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: X1
metadata:
 name: x1-release
spec:
 properties:
 api:
 samlEnabled: true
 samlWebSsoEndpoint: https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/exkexl6xc9MhzqiC30h7/sso/saml
 samlEntityId: http://www.okta.com/exkfdsn6oy5xywqC00h7
 samlRegistrationId: simplesamlphp
 samlSpEntityId: turbo
 samlIdpCertificate: |
 -----BEGIN CERTIFICATE-----
```



```

MIIDpDCCAoygAwIBAgIGAWMnhv7cMA0GCSqGSIb3DQEBQwUAMIGSMQswCQYDVQQGEwJVUzETMBEG
A1UECAwKQ2FsaWZvcmlkZXRlcjEzARBgNVBAMMcmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
MBIGA1UECwwLU1NPUHJvdmlkZXRlcjEzARBgNVBAMMcmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
DwluZm9Ab2t0YS5jb20wHhcNMTgwNTAzMTk0MTI4WhcNMjgwNTAzMTk0MjI4WjCBKjELMAkGA1UE
BhMCCVVMxEzARBgNVBAGMCKNhbgG1mb3JuaWExFjAUBGNVBAcMDVNBhbiBGcmFuY2IzY28xDTALBgNV
BAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRMwEQYDVQDDApkZXYtNzcwMjAyMRwwGgYj
KoZiIhvcNAQkBFg1pbmZvQG9rdGExY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
ugxQGqHAXpjVQZws09n818bFCoEevH3AZbz7568XuQm6MK6h7/09wB4C5oUYddemt5t2Kc8GRhf3
BDXX5MVZ8G9AUUpG1MSqe1CLV2J96rMnwMIJsKeRXr01LYxv/J4kjkntpOC389wmcy2fE4RbPoJne
P4u2b32c2/V7xsJ7UEjPPSD4i812QG6qsUkkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
PTpzek8NS5M5rvF8yaD+eE1zS0I/HicHbPOVvLa10JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
nn4XW9Cqjk3equSCJPo5o1M5j8v1LrJYVarqhwIDAQAQBA0GCSqGSIb3DQEBQwUAA4IBAQC26kYe
LgqjIkF5rvxB2QzTgcd0LVzX0uiVVTZr8Sh5714jJqbDoIgvaQQrxRSQzD/X+hcmhuwdp9s8zPHS
JagtUJXiypwNtrzbif6M71trWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQoOIVIjHx9wgynaAK
HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70E0wp
6Ta1uRlQh6jCCOCWRuZbbz2T3/s0X+sibC4rLlIwfyTkcUopF/bTSDwWknoRskK4dBekFcvN9N+C
p/qaHYcQd6i2vyor888DLHDPXhSKWhpG
-----END CERTIFICATE-----

```

6. Save your changes to the CR file.

7. Apply the modified cr.yaml file.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

8. Restart the API component to load the new spec.

a. Open an SSH terminal session to your Turbonomic instance.

b. Restart the API component.

```
kubectl delete pod api- <API_POD_ID> (NOTE: To auto-fill the pod ID, type api- and then press TAB.)
```

9. Verify that the configuration is successful.

a. Navigate to the Turbonomic User Interface.

You will be automatically redirected to your IdP for authentication.

b. Log in with the username that is a member of the external group or external user that you previously configured.

c. Verify that the system time on your Turbonomic instance is correct.

If the time is not synchronized, this might cause an HTTP Status 401 - authentication failed exception in the browser.

d. If the configuration is not successful, look for an HTTP Status 500 exception in the product log. If this exception exists, review your CR file for invalid entries.

## Example of IdP Metadata

This section provides an example of IdP metadata which may be useful when you are examining the optional attributes in your metadata.

If your metadata includes optional attribute tags that are not listed in the example, remove those optional attribute tags since they are not supported.

```

<?xml version="1.0" encoding="UTF-8"?>
 <md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
 entityID="http://www.okta.com/exkexl6xc9MhzqiC30h7">
 <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
 <md:KeyDescriptor use="signing">
 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
 <ds:X509Data>
 <ds:X509Certificate>
MIIDpDCCAoygAwIBAgIGAWMnhv7cMA0GCSqGSIb3DQEBQwUAMIGSMQswCQYDVQQGEwJVUzETMBEG
A1UECAwKQ2FsaWZvcmlkZXRlcjEzARBgNVBAMMcmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
MBIGA1UECwwLU1NPUHJvdmlkZXRlcjEzARBgNVBAMMcmRldi03NzEyMDIxHDAaBgkqhkiG9w0BCQEW
DwluZm9Ab2t0YS5jb20wHhcNMTgwNTAzMTk0MTI4WhcNMjgwNTAzMTk0MjI4WjCBKjELMAkGA1UE
BhMCCVVMxEzARBgNVBAGMCKNhbgG1mb3JuaWExFjAUBGNVBAcMDVNBhbiBGcmFuY2IzY28xDTALBgNV
BAoMBE9rdGExFDASBgNVBAsMC1NTT1Byb3ZpZGVyMRMwEQYDVQDDApkZXYtNzcwMjAyMRwwGgYj
KoZiIhvcNAQkBFg1pbmZvQG9rdGExY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
ugxQGqHAXpjVQZws09n818bFCoEevH3AZbz7568XuQm6MK6h7/09wB4C5oUYddemt5t2Kc8GRhf3
BDXX5MVZ8G9AUUpG1MSqe1CLV2J96rMnwMIJsKeRXr01LYxv/J4kjkntpOC389wmcy2fE4RbPoJne
P4u2b32c2/V7xsJ7UEjPPSD4i812QG6qsUkkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
PTpzek8NS5M5rvF8yaD+eE1zS0I/HicHbPOVvLa10JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
nn4XW9Cqjk3equSCJPo5o1M5j8v1LrJYVarqhwIDAQAQBA0GCSqGSIb3DQEBQwUAA4IBAQC26kYe
LgqjIkF5rvxB2QzTgcd0LVzX0uiVVTZr8Sh5714jJqbDoIgvaQQrxRSQzD/X+hcmhuwdp9s8zPHS
JagtUJXiypwNtrzbif6M71trWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQoOIVIjHx9wgynaAK
HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70E0wp
6Ta1uRlQh6jCCOCWRuZbbz2T3/s0X+sibC4rLlIwfyTkcUopF/bTSDwWknoRskK4dBekFcvN9N+C
p/qaHYcQd6i2vyor888DLHDPXhSKWhpG

```

```

KoZIhvcNAQkBFg1pbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA
ugxQGqHAXpjvQZws09n8l8bFcoEevH3AZbz7568XuQm6MK6h7/09wB4C5oUYddemt5t2Kc8GRhf3
BDXX5MVZ8G9AUpG1MSqe1CLV2J96rMnwMIJsKeRXr01LYxv/J4kjkntpOC389wmcY2fE4RbPoJne
P4u2b32c2/V7xsJ7UEjPPSD4i8l2QG6qsUkx3AyNsjo89PekMfm+Iu/dFKXkdjwXZXPxaL0HrNW
PTpezk8NS5M5rvF8yaD+eE1zS0I/HicHbP0VvLa10JZyN/f4bp0XJkxZJz6jF5DvBkwIs8/Lz5GK
nn4W9Cqjk3equSCJPo5o1Msj8v1LrJYVarqhwIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQC26kYe
LgqjIkF5rvxB2QzTgcd0LVzX0uiVVTZr8Sh5714jJqbDoIgvaQQrxRSQzD/X+hcmhuwdp9s8zPHS
JagtUJXiypwNtrzbF6M71trWB9sdNrqc99d1gOVRr0Kt5pLTaLe5kkq7dRaQo0IVIJhX9wgynaAK
HF/SL3mHUytjXggs88AAQa8JH9hEpwG2srN8EsizX6xwQ/p92hM2oLvK5CSMwTx4VBuGod70E0wp
6Ta1uRLQh6jCC0CWruZbbz2T3/s0X+sibC4rLI1wfyTkcUopF/bTsdWwknORskK4dBekFcvN9N+C
p/qaHYcQd6i2vyor888DLHDPXhSKWhpG
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</md:KeyDescriptor>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/
exkexl6xc9MhzqiC30h7/sso/saml"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://dev-771202.oktapreview.com/app/ibmdev771202_turbo2_1/
exkexl6xc9MhzqiC30h7/sso/saml"/>
</md:IDPSSODescriptor>
</md:EntityDescriptor>

```

## Setting Up OpenID Authentication

According to the OpenID Foundation, "OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol". OpenID Connect enables clients to verify user identity via a given authentication server. Turbonomic supports OpenID authentication through the following providers:

- Google
- IBM-MCM
- Okta

### Logging in to Turbonomic with OpenID

When you configure OpenID in Turbonomic, the platform registers the OpenID clients that you specify. To log in through of these OpenID clients, you manually navigate to a URL that tells Turbonomic which client to use. It then redirects to the OpenID login screen for that given client.

The URL that you provide is in the form:

```
https://{hostname}/vmturbo/oauth2/login/code/{openIdClients}
```

Where:

- `{hostname}` is the host address for your installation of Turbonomic
- `{openIdClients}` is the is the client name of the OpenID provider you want to use

You specify this as the `openIdClients` property when you configure OpenID in Turbo.

#### Note:

This URL must also be set in the "Authorized direct URIs" section of your provider's OpenID configuration. For example, assume your Turbonomic host address is 10.10.12.34, and you configured an Okta OpenID client. In that case, when the Turbonomic login screen appears, you would navigate to:

```
https://10.10.12.34/vmturbo/oauth2/login/code/okta
```

After you navigate to that URL, the browser redirects to the OpenID login screen, where you can enter credentials for a single user or a user group.

#### Note:

To authenticate a user group, the group must be configured on the OpenID provider, and also on Turbonomic. The group name must be identical in both configurations.

On the OpenID provider, the client that you are using should include groups scope values that give specific names for user groups. Contact your OpenID administrator to get the group names. Then, on Turbonomic you should create user groups that use the same names.

For example, assume the OpenID ID token includes the following groups claim:

```
{
 "sub": "1234567890",
 "name": "My_User_Name",
 "iat": "12121212",
 "groups": "My_Special_User_Group"
}
```

To use the group `My_Special_User_Group` for authentication, you must create a user group in Turbonomic with the name `My_Special_User_Group`. Any members of that group will then get the role you have assigned to that user group.

## Configuring OpenID on Turbonomic

To configure Turbonomic to authenticate via OpenID:

1. (Required) Ensure that `chrony` is configured and the system time on your Turbonomic instance is correct.

For instructions, see [Synchronizing Time](#).

2. Obtain the necessary data from your OpenID provider.

Contact your security administrator to obtain the data from the provider. You will use this data to configure SSO in the Turbonomic CR file located at:

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

The data you need and the properties you declare in the CR file will differ depending on the OpenID provider you want to use:

- Google:

| CR Fields:                      | Description:                                                                                  |
|---------------------------------|-----------------------------------------------------------------------------------------------|
| <code>openIdClients</code>      | <code>google</code><br>The name of the OpenID client you are using to perform authentication. |
| <code>openIdClientId</code>     | The OAuth2 Client Identifier for the OpenID client that you are using.                        |
| <code>openIdClientSecret</code> | The OAuth2 Client Secret for the OpenID client that you are using.                            |

- IBM-MCM:

| CR Fields:                 | Description:                                                                               |
|----------------------------|--------------------------------------------------------------------------------------------|
| <code>openIdClients</code> | <code>ibm</code><br>The name of the OpenID client you are using to perform authentication. |

| <b>CR Fields:</b>          | <b>Description:</b>                                                    |
|----------------------------|------------------------------------------------------------------------|
| openIdClientAuthentication | post<br>The client authentication method.                              |
| openIdUserAuthentication   | form<br>The user authentication method.                                |
| openIdClientId             | The OAuth2 Client Identifier for the OpenID client that you are using. |
| openIdClientSecret         | The OAuth2 Client Secret for the OpenID client that you are using.     |
| openIdAccessTokenUri       | The URI the login process will use to get an Access Token.             |
| openIdUserAuthorizationUri | The URI to the Authorization Endpoint for OpenID Connect.              |
| openIdUserInfoUri          | The URI to the OpenID Connect UserInfo endpoint.                       |
| openIdJwkSetUri            | The URI to get the JSON Web Key set that can verify the Access Token.  |
| openIdExternalGroupTag     | The name of a custom group to use for authentication.                  |

- Okta:

| <b>CR Fields:</b>          | <b>Description:</b>                                                            |
|----------------------------|--------------------------------------------------------------------------------|
| openIdClients              | okta<br>The name of the OpenID client you are using to perform authentication. |
| openIdClientId             | The OAuth2 Client Identifier for the OpenID client that you are using.         |
| openIdClientSecret         | The OAuth2 Client Secret for the OpenID client that you are using.             |
| openIdAccessTokenUri       | The URI the login process will use to get an Access Token.                     |
| openIdUserAuthorizationUri | The URI to the Authorization Endpoint for OpenID Connect.                      |
| openIdUserInfoUri          | The URI to the OpenID Connect UserInfo endpoint.                               |
| openIdJwkSetUri            | The URI to get the JSON Web Key set that can verify the Access Token.          |

3. Update the Turbonomic CR file with your configuration data.

You now have the data that you need to configure SSO via OpenID. You will edit the `cr.yaml` file that configures your Turbonomic node, and then deploy or restart the node.

- Open the CR file for editing.

In a shell, cd to the deploy/crds directory in the Turbonomic VM:

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```

Then open the CR file for editing. For example, to open the file in VI:

```
vi charts_v1alpha1_xl_cr.yaml
```

As you edit this file, you will refer to the dat that you obtained from your authentication provider.

- In the CR file, navigate to the entry for the API component.

In the CR file search for or scroll to the entry:

```
apiVersion: charts.helm.k8s.io/v1alpha1
```

You will make changes to this component spec, under spec:properties:api:

- Turn on the OpenID feature.

For the first API property, set the following:

```
openIdEnabled: true
```

The file should be similar to:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
 name: xl-release
spec:
 properties:
 api:
 openIdEnabled: true
```

- Enter the relevant OpenId data for your authentication provider. The CR file should be similar to these examples, depending on which provider you use:

- Google:

The file should be similar to:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
 name: xl-release
spec:
 properties:
 api:
 openIdEnabled: true
 openIdClients: google
 openIdClientId: xxxx-4vinrdgllag5p84jjebc6xxxxxx5u.apps.googleusercontent.com
 openIdClientSecret: xxxxxhGcdFEjQa-xxxxxx
```

- IBM-MCM:

The file should be similar to:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
 name: xl-release
spec:
 properties:
 api:
 openIdEnabled: true
 openIdClients: ibm
 openIdClientAuthentication: post
 openIdUserAuthentication: form
 openIdClientId: turbonomic-mcm-demo
```

```
openIdClientSecret: "xxxxxxvZ2ZscDht0FVxxxxxxU3d6cXR4cTZhb2xxxxxxRT0K"
openIdAccessTokenUri: https://icp-console.apps.blue-13.dev.multicloudops.io/
idprovider/v1/auth/token
openIdUserAuthorizationUri: https://icp-console.apps.blue-13.dev.multicloudops.io/
idprovider/v1/auth/authorize
openIdUserInfoUri: https://icp-console.apps.blue-13.dev.multicloudops.io/v1/auth/
userinfo
openIdJwkSetUri: https://icp-console.apps.blue-13.dev.multicloudops.io/oidc/
endpoint/OP/jwk
```

– Okta

The file should be similar to:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: X1
metadata:
 name: x1-release
spec:
 properties:
 api:
 openIdEnabled: true
 openIdClients: okta
 openIdClientId: xxxxxxxxh1xhQnSKxxxx
 openIdClientSecret: xxxxxxxxxtIhVCIRUnhq4xxxxxxDdhLdqx0
 openIdAccessTokenUri: https://vmturbo.okta.com/oauth2/v1/token
 openIdUserAuthorizationUri: https://vmturbo.okta.com/oauth2/v1/authorize
 openIdUserInfoUri: https://vmturbo.okta.com/oauth2/v1/userinfo
 openIdJwkSetUri: https://vmturbo.okta.com/oauth2/v1/keys
```

4. Save your changes to the CR file.
5. Apply the modified cr.yaml file.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

6. Restart the API component to load the new spec.
  - a. Open an SSH terminal session to your Turbonomic instance.
  - b. Restart the API component.

```
kubectl delete pod api-<API_POD_ID> (NOTE: To auto-fill the pod ID, type api- and then press TAB.)
```
7. Verify that the configuration is successful.
  - a. Navigate to the Turbonomic User Interface.

You will be automatically redirected to your authentication provider for authentication.
  - b. Log in with the username that is a member of the external group or external user that you previously configured.
  - c. Verify that the system time on your Turbonomic instance is correct.

If the time is not synchronized, this might cause an HTTP Status 401 -authentication failed exception in the browser.
  - d. If the configuration is not successful, look for an HTTP Status 500 exception in the product log. If this exception exists, review your CR file for invalid entries.

## Disabling Single Sign-On

If for some reason you no longer want to use SSO, you can disable it for your Turbonomic installation. To disable Single Sign-On, perform these steps:

1. Update the SSO configuration to disable it.
  - a. Open an SSH terminal session to your Turbonomic instance.

b. Open the CR file for editing.

In a shell, cd to the `deploy/crds` directory in the Turbonomic VM:

```
cd /opt/turbonomic/kubernetes/operator/deploy/crds
```

Then open the CR file for editing. For example, to open the file in VI:

```
vi charts_v1alpha1_xl_cr.yaml
```

c. In the CR file, navigate to the entry for the API component.

In the CR file search for or scroll to the entry:

```
apiVersion: charts.helm.k8s.io/v1alpha1
```

You will make changes to this component spec, under `spec:properties:api:`

d. Turn off the SSO feature.

The entry to set to false is different depending on whether you use SAML or OpenID authentication:

- SAML Authentication:

Find the `samlEnabled` property to false. It should appear as follows:

```
samlEnabled: false
```

- OpenID Authentication:

Find the `openIdEnabled` property to false. It should appear as follows:

```
openIdEnabled: false
```

e. Save your changes to the CR file.

2. Apply the modified `cr.yaml` file.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

3. Restart the API component.

In the same SSH terminal session that you opened to edit the CR file:

a. Use `sudo` as root.

```
sudo bash
```

b. Restart your API component.

```
kubectl delete pod api-<API_POD_ID>
```

**Note:**

To auto-fill the pod ID, type `api-` and then press TAB.

4. Verify that the configuration is successful.

a. Navigate to the Turbonomic User Interface.

You will no longer be redirected to your IdP for authentication. You will be redirected to the default Turbonomic login screen.

b. Log in with a local account or an Active Directory (AD) account.

## Updating Turbonomic to a New Version

---

### Note:

If you are updating from Turbonomic version 7.21.x, or 7.17.x, contact your Technical Support representative.

Starting with Turbonomic version 8.7.5, IBM Container Registry is used for all Turbonomic images for online upgrades. Refer to your preferred update method for more information.

IBM continually and rapidly innovates and improves all aspects of this product. IBM releases newer versions of this product every two weeks. You should check regularly to see if a new version is available by visiting the [IBM Turbonomic documentation site](#).

When a new version is available, it is important to properly update your existing installed instance. When you first installed Turbonomic, you put into place sophisticated data collection and analysis processes, and your database retains performance data from across your virtual environment. Turbonomic uses this historical data for right-sizing, projecting trends, and other analysis. This means that the database is important to Turbonomic and becomes more so over time. Properly updating your installation of Turbonomic preserves the database for continued use.

### OVA updates

Before you begin the update procedure:

- Review [What's New](#) and [Fixed Issues](#) to see what is new for this release.

#### Note:

As Turbonomic evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

- Make sure you have the email that IBM sent to you with links to the Turbonomic OVA file and to the ISO image.
- For on-prem installations, make sure that the physical machine hosting the VM meets the minimum requirements (see [Minimum Requirements](#)).
- Ensure you are running the correct version of the historical database.

For its default historical database, Turbonomic currently supports MariaDB version 10.5.20. This support includes comprehensive testing and quality control for Turbonomic usage of the historical database.

For more information, see [Verifying your MariaDB Version](#).

- Execute the `upgrade-precheck.sh` script.

You can use this script to make sure that your current installation of Turbonomic is ready to update. We strongly recommend that you run this script before going on to update your installation (see [Checking Before Updating](#)).

Update your Turbonomic VM using either of the following methods:

- Online method, if you have access to the internet:

See [Online Update](#)

- Offline method, via a downloaded ISO image:

See [Offline Update](#)



## Red Hat OpenShift updates

- If you are running Turbonomic on Red Hat OpenShift, see [Updating using the Red Hat OpenShift Operator Hub](#).

## Checking Before Updating

Before you perform an update of your Turbonomic instance, you should execute the script, `upgrade-precheck.sh`. This script inspects your installation to check for the following:

- Sufficient free disk space
- For online updates, access to required endpoints (`icr.io`, `github.com`, etc.)
- The MariaDB service is running

Note that this check is for the default installation of the MariaDB service, only. For example, the script does not check an external installation of MySQL or MariaDB, if that is the historical database you have configured. In that case, the script will indicate that your MariaDB service is not running. For an external database deployment, this is a normal result.

- The Kubernetes service is running
- The necessary Kubernetes certificates are valid

If the certificates are not valid, you can run the `kubeNodeCertUpdate.sh` script to correct the issue. This script should be located on your installation at `/opt/local/bin`. For more information, contact your support representative.

- Root password is not set to expire
- Time sync is enabled, and current if running
- All Turbonomic pods are running

To execute this script:

1. Download the latest version of the script.

- a. Log in to the Turbonomic VM.

Use SSH to log in to the Turbonomic VM using the `turbo` account and password.

- b. Change to the scripts directory.

```
cd /opt/local/bin
```

- c. Get the latest version of the script.

```
curl -O --proxy PROXY_NAME_IP:PORT \
https://\
raw.githubusercontent.com/turbonomic/t8c-install/master/bin/upgrade-precheck.sh
```

Where `--proxy PROXY_NAME_IP:PORT` is an optional specification to execute the download through a proxy.

- d. Make the script executable.

```
chmod +x upgrade-precheck.sh
```

2. Execute the script.

```
./upgrade-precheck.sh
```

As the script executes, it identifies any issues that you should address before you execute an update.

## External DBs and Turbonomic Updates

If you have deployed Turbonomic with an external database server, for some updates you might need to manually create a new database and user for that deployment. This is important if your external database server is multi-tenant, or if your deployment does not grant administrative privileges to Turbonomic.

### Note:

*If your external database server is multi-tenant, or if your database server does not grant administrative privileges to Turbonomic, then you must continue with this configuration requirement.*

Azure database services are multi-tenant. If you deployed an external database on Azure, this configuration requirement applies to you.

If you deployed your database server in a way that grants Turbonomic privileges to create new databases and new users, then a product update will automatically create the required database. This configuration requirement does not apply to you and you do not need to take any action.

For some Turbonomic updates, the updated version includes new databases on the historical database server. If you are updating to one of these versions, then you must *first* create the new database, and a user account with privileges to access that database.

This table lists the Turbonomic versions that require new databases. If you are updating from a version earlier than one of these, you must create the indicated new databases. For example, if you are updating from version 8.0.0 to 8.2.0, then you must create the `repository` database and the `market` database.

| <b>Turbonomic Version:</b> | <b>New Databases:</b>   | <b>Notes:</b>                                                                                                                                                        |
|----------------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8.2.5                      | <code>api</code>        | If you are updating from a version earlier than 8.2.5, you must create a new database named <code>api</code> , and a user account named <code>api</code> .           |
| 8.1.1                      | <code>market</code>     | If you are updating from a version earlier than 8.1.1, you must create a new database named <code>market</code> , and a user account named <code>market</code> .     |
| 8.0.1                      | <code>repository</code> | If you are updating from a version earlier than 8.0.1, you must create a database named <code>repository</code> , and a user account named <code>repository</code> . |

### Note:

If you have already updated to one of these versions of Turbonomic, and you did not perform the steps to update your external DB, please contact your support representative.

To create the databases and users, you will:

- Manually create each required database

This includes creating the database in your DB instance, creating a user to access the database, and granting privileges to the user.

- Manually add the each required database to your `cr.yaml` file

The `cr.yaml` file declares entries for each component database. Each entry names the component, and gives the user and password that the component can use to access that database. You must add a new entry for each new database.

To create a new database:

1. Connect to your external DB using a global account.

The account must have privileges to create databases and users. If you have specified `dbRootUsername` in the `cr.yaml` file, you can use that account.

2. Create the database, where **<New\_Database>** matches the database name in the table above:

```
create database <New_Database>;
```

For example, to create a new `api` database, execute:

```
create database api;
```

3. Create the account that Turbonomic will use to access the database where **<New\_Database>** matches the database name in the table above:

```
create user '<New_Database>'@'%' identified by 'vmturbo';
```

For example, to create a user for the `api` database, execute:

```
create user 'api'@'%' identified by 'vmturbo';
```

**Note:**

The value **vmturbo** is the default password that Turbonomic uses for all component database accounts. If you have manually created accounts with different credentials, you can do so for this database as well.

4. Set the user account privileges for the new user account, where **<New\_Database>** matches the database name in the table above:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER
ROUTINE, EVENT, TRIGGER ON <New_Database>.* TO '<New_Database>'@'%' ;
```

For example, to set account privileges for the `api` user, execute:

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, INDEX, ALTER, CREATE
TEMPORARY TABLES, LOCK TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, CREATE ROUTINE, ALTER
ROUTINE, EVENT, TRIGGER ON api.* TO 'api'@'%' ;
```

5. Flush privileges to make the privileges take effect:

```
flush privileges;
```

Now that the new database is created in your external DB service, you must declare access to it the Turbonomic `cr.yaml` resource.

1. Open the `.cr` file for editing. The location of the file depends on the type of Turbonomic installation you are configuring.

| <b>VM Image installation of Turbonomic:</b>                                                                                                                                                                                                                  | <b>Turbonomic on a Kubernetes node or node cluster:</b>                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| Open a SSH terminal session on your Turbonomic instance<br>Log in with the System Administrator that you set up when you installed Turbonomic: <ul style="list-style-type: none"><li>• Username: turbo</li><li>• Password: [your_private_password]</li></ul> | Open the following file for editing:<br><pre>deploy/crds/charts_v1alpha1_x1_cr.yaml</pre> |

|                                                                                                       |                                                         |
|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------|
| <b>VM Image installation of Turbonomic:</b>                                                           | <b>Turbonomic on a Kubernetes node or node cluster:</b> |
| Then edit the file:<br><br>/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml |                                                         |

2. Add the credentials for the matching pod to access the new database.

Add the entry to the `properties:` section of the `cr.yaml` file, where **vmturbo** is the password that you assigned to that user account, **yourDB** is the qualified name of your external DB or your multi-tenant DB partition, and **<New\_Database>** is the name of your new database. Declare the following in the entry you add:

```
<New_Database>:
 <New_Database>DbUsername: <New_Database>@yourDB
 <New_Database>DbPassword: vmturbo
```

For example, if you added the `api` database, the resulting `cr.yaml` file should be similar to:

```
properties:
 global:
 enableSecureDBConnection: true
 sqlDialect: MYSQL
 dbRootPassword: yourAdminPassword
 dbRootUsername: xladmin@yourDB
 #dbUserPassword:
 #dbUsername:
 action-orchestrator:
 actionDbUsername: action@yourDB
 actionDbPassword: yourPassword
 auth:
 authDbUsername: auth@yourDB
 authDbPassword: yourPassword
 clustermgr:
 clustermgrDbUsername: clustermgr@yourDB
 clustermgrDbPassword: yourPassword
 cost:
 costDbUsername: cost@yourDB
 costDbPassword: yourPassword
 group:
 groupComponentDbUsername: group_component@yourDB
 groupComponentDbPassword: yourPassword
 history:
 historyDbUsername: history@yourDB
 historyDbPassword: yourPassword
 plan-orchestrator:
 planDbUsername: plan@yourDB
 planDbPassword: yourPassword
 topology-processor:
 topologyProcessorDbUsername: topology_processor@yourDB
 topologyProcessorDbPassword: yourPassword
 repository:
 repositoryDbUsername: repository@yourDB
 repositoryDbPassword: yourPassword
 market:
 marketDbUsername: market@yourDB
 marketDbPassword: yourPassword
 api:
 apiDbUsername: api@yourDB
 apiDbPassword: yourPassword
```

After you have done this, you can update to the latest version of Turbonomic. (Note that upgrading applies changes to the version information in this file.)

## Online Update

Online updates assume that you have direct access to the Internet or access to the Internet through a proxy server.

## Connection Requirements

If you are installing from behind a firewall, make sure you have full access to the IBM Container Registry services that deliver the Turbonomic components. IBM Container Registry is a publicly accessible registry that does not require login credentials. Addressing for these services can be different depending on details of your current environment. Note that IBM has no control over these endpoints.

### Note:

Starting with Turbonomic version 8.7.5, IBM Container Registry is used for all Turbonomic images for online upgrades and new installs. Ensure you have access to `https://icr.io` before continuing.

As of this writing, you need access to the following:

- `https://icr.io`
- `https://github.com`
- `https://raw.githubusercontent.com`
- `https://download.vmturbo.com`
- `https://yum.mariadb.org`
- `http://archive.mariadb.org/`
- `https://packagecloud.io`
- `https://download.postgresql.org`
- `https://yum.postgresql.org`

If your environment supports access through a proxy, you can set up the containerd daemon to use that proxy via the `HTTP_PROXY`, `HTTPS_PROXY`, and `NO_PROXY` environment variables. Create the file:

```
/etc/systemd/system/containerd.service.d/http-proxy.conf
```

Configure the proxies in the `[Service]` section of the file, where `NO_PROXY` specifies hosts to exclude from proxying:

```
[Service]
Environment="HTTP_PROXY=http://proxy.example.com:80"
Environment="HTTPS_PROXY=https://proxy.example.com:443"
Environment="NO_PROXY=localhost,127.0.0.1,containerd-registry.example.com,.corp"
```

Then reload the daemon and restart containerd:

```
sudo systemctl daemon-reload
sudo systemctl restart containerd
```

For more information, see [Container runtime changes in Kubernetes 1.24 and beyond](#) on the Kubernetes documentation site.

If none of these options work, then you should perform an offline update.

## Performing an Online Update

### Note:

If you are experiencing issues performing the update, contact your Technical Support representative.

To perform an online update of your Turbonomic installation:

1. Download the current update script.

You must download the update script that matches the version to which you are updating.

**Note:**

Scripted online updates support updating to version 8.0.6 or later.

- a. Log in to the Turbonomic VM.

Use SSH to log in to the Turbonomic VM using the turbo account and password.

- b. Change to the scripts directory.

```
cd /opt/local/bin
```

- c. Get the script for your update version. Run the following command as a single line command:

```
curl -O --proxy PROXY_NAME_IP:PORT \
https://download.vmturbo.com/appliance/download/updates/8.9.3/onlineUpgrade.sh
```

Where `--proxy PROXY_NAME_IP:PORT` is an optional specification to execute the download through a proxy.

- d. Make the script executable.

```
chmod +x onlineUpgrade.sh
```

2. Save a snapshot of your current Turbonomic VM.

Before updating, shut down (not power off) the Turbonomic VM.

```
sudo init 0
```

Then, perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.

3. Optionally, enable new probes in your environment.

**Note:**

As Turbonomic evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

For steps to enable new probes in your updated version, see [Enabling and Disabling Probe Components](#). Use these steps to edit the platform's `cr.yaml` file.

**Note:**

During the online or offline upgrade process, you should not apply these changes at this time.

4. Start the online update by running the update script.

```
./onlineUpgrade.sh 8.9.3
```

**Note:**

Ensure the version in the online update script matches the version of the update script that you downloaded. Also, you cannot use this script to move to an earlier version.

Wait until the script completes its execution.

5. Verify that the Turbonomic application installed correctly.

```
kubectl get pods -n turbonomic
```

After all of the pods start, the READY column should read 1/1, 2/2, and so on and the STATUS column should read Running for each pod.

**Note:**

Under rare circumstances the downloads from Docker Hub can fail. This occurs when the limit on image downloads has been exceeded. As a result, one or more pods can show ImagePullBackOff as the pod STATUS. Note that this does not occur when using the IBM Container Registry as it does not have download limits.

If you encounter this STATUS, you can wait until the limitation period expires. At that time, the download of pods will continue, and the platform should start up normally.

6. Verify that you are running the correct version of MariaDB.

For this version of the product, Turbonomic supports MariaDB version 10.5.20. Even after updating to this Turbonomic version, it is possible that your installation is running an earlier version of MariaDB.

While still in the SSH session, check the MariaDB version:

```
mysql -u root --password=my_pwd -e "SHOW VARIABLES LIKE 'version';"
```

The output should be similar to:

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| version | 10.5.20-MariaDB |
+-----+-----+
```

If the MariaDB version is earlier than 10.5.20, you must update your MariaDB. For complete instructions and information, see [Verifying your MariaDB Version](#).

7. Clear your browser data and refresh your browser.

After clearing the browser data and refreshing your browser, you have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

8. Notify other users to clear their browser data and refresh their Turbonomic browser sessions.

## Offline Update

To perform an offline update of your Turbonomic installation:

1. Save a snapshot of your current Turbonomic VM.

Before updating, shut down (not power off) the Turbonomic VM.

```
sudo init 0
```

Then, perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.

2. Optionally, enable new probes in your environment.

**Note:**

As Turbonomic evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

For steps to enable new probes in your updated version, see [Enabling and Disabling Probe Components](#). Use these steps to edit the platform's cr.yaml file.

**Note:**

During the online or offline upgrade process, you should not apply these changes at this time.

3. Download the ISO image.

Refer to the email you received from IBM for links to the Turbonomic OVA file and to the ISO image.

4. Save the ISO image to a location that is available to the VM that runs Turbonomic. Then mount the image as a CD drive.

For example, if you run the Turbonomic VM in vCenter Server do the following:

- a. In vCenter, navigate to the Turbonomic VM.
- b. Right-click the VM and choose **Edit Settings**.
- c. In the CD/DVD Drive drop-down menu, select **Datastore ISO**, then browse to the Turbonomic update ISO image and choose it.
- d. Ensure that **Connect at power on** is selected.

5. Open an SSH terminal session to your Turbonomic instance.

After you make a snapshot or clone of your current Turbonomic VM, open an SSH session. Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Username: [your\_private\_password]

6. Get the script for your update version.

On your local machine, navigate to where you want to download the script. Run the following command as a single line command:

```
curl -O --proxy PROXY_NAME_IP:PORT \
https://download.vmturbo.com/appliance/download/updates/{VNUM}/isoUpdate.sh
```

Where

- `--proxy PROXY_NAME_IP:PORT` is an optional specification to execute the download through a proxy.
- `{VNUM}` is the version you are updating to. The version must be 8.2.5 or later.

7. Upload the script to your Turbonomic instance.

Execute a file transfer from your local machine to the Turbonomic server. Save the script to `/opt/local/bin/` on the VM that runs Turbonomic.

8. Make the script executable.

```
chmod +x /opt/local/bin/isoUpdate.sh
```

9. Execute the offline installation.

```
/opt/local/bin/isoUpdate.sh
```

As the script executes, it:

- Backs up the old scripts in your installation.
- Updates the configuration and code assets in your installation.
- Updates the platform to the new version.
- Updates custom resources.
- Updates the MariaDB configuration (but this does not update the MariaDB version).



- If you have enabled Embedded Reports or Data Export, installs the Embedded Reports and Data Export database (Postgres and TimescaleDB).
- Scales down the t8c-operator and the Turbonomic components.
- Executes the final updates for this version.
- Scales up the t8c-operator, which then restarts the Turbonomic components.

After the script is finished updating your platform, you should give it enough time for all the components to start up again.

#### 10. Verify that the Turbonomic application installed correctly.

```
kubectl get pods -n turbonomic
```

After all of the pods start, the READY column should read 1/1, 2/2, and so on and the STATUS column should read Running for each pod.

You should see output similar to the following:

| NAME                                       | READY | STATUS  | RESTARTS |
|--------------------------------------------|-------|---------|----------|
| action-orchestrator-b6454c9c8-mfl85        | 1/1   | Running | 0        |
| api-7887c66f4b-shndq                       | 1/1   | Running | 0        |
| arangodb-7f646fc5fc-zhcwf                  | 1/1   | Running | 0        |
| auth-5b86976bc8-vxwz4                      | 1/1   | Running | 0        |
| clustermgr-85548678d9-r5wb8                | 1/1   | Running | 0        |
| consul-7f684d8cb8-6r677                    | 1/1   | Running | 0        |
| cost-5f46dd66c4-6d6cb                      | 1/1   | Running | 0        |
| extractor-5f41dd61c4-4d6lq                 | 1/1   | Running | 0        |
| group-5bfdfbc6f8-96bsp                     | 1/1   | Running | 0        |
| history-5fc7fbc855-6zslq                   | 1/1   | Running | 0        |
| kafka-74cc77db94-dfrbl                     | 1/1   | Running | 0        |
| market-5f54699447-z4wkm                    | 1/1   | Running | 0        |
| mediation-actionscript-57b4fc6df-4lzfz     | 1/1   | Running | 0        |
| mediation-appdynamics-6d65f8766f-kb441     | 1/1   | Running | 0        |
| mediation-hpe3par-d7c475c4c-v8ftc          | 1/1   | Running | 0        |
| mediation-hyperv-6bd8c94df5-4dbzx          | 1/1   | Running | 0        |
| mediation-netapp-7f8fc955d9-4kkdl          | 1/1   | Running | 0        |
| mediation-oneview-7dbd7b54cf-7rfqp         | 1/1   | Running | 0        |
| mediation-pure-58c4bd8cd9-8n256            | 1/1   | Running | 0        |
| mediation-ucs-6f4bb9889-9rnqk              | 1/1   | Running | 0        |
| mediation-vcenter-5bc4f5fbd4-nzm4j         | 1/1   | Running | 0        |
| mediation-vcenterbrowsing-5c5987f66c-bfjq4 | 1/1   | Running | 0        |
| mediation-vmx-6c59969b89-28t9j             | 1/1   | Running | 0        |
| mediation-vmw-9c4878cf9-rfxnl              | 1/1   | Running | 0        |
| nginx-5b775f498-sm2mm                      | 1/1   | Running | 0        |
| plan-orchestrator-6dffc4c9b6-p5t5n         | 1/1   | Running | 0        |
| reporting-b44fbd4-8fjv5                    | 1/1   | Running | 0        |
| repository-6d555bb4bf-fxldh                | 1/1   | Running | 0        |
| rsyslog-fd694878c-5tb2c                    | 1/1   | Running | 0        |
| t8c-operator-558bcc758d-5h8mp              | 1/1   | Running | 0        |
| topology-processor-b646b786b-9skp7         | 1/1   | Running | 0        |
| zookeeper-5f65b5bf69-nnmbt                 | 1/1   | Running | 0        |

#### 11. Verify that you are running the correct version of MariaDB.

For this version of the product, Turbonomic supports MariaDB version 10.5.20. Even after updating to this Turbonomic version, it is possible that your installation is running an earlier version of MariaDB.

While still in the SSH session, check the MariaDB version.

```
mysql -u root --password=my_pwd -e "SHOW VARIABLES LIKE 'version';"
```

The output should be similar to:

```
+-----+
| Variable_name | Value |
+-----+
| version | 10.5.20-MariaDB |
+-----+
```

If the MariaDB version is earlier than 10.5.20, you should update your MariaDB. For complete instructions and information, see [Verifying your MariaDB Version](#).

12. Clear your browser data and refresh your browser.

After clearing the browser data and refreshing your browser, you have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

13. Notify other users to clear their browser data and refresh their Turbonomic browser sessions.

## Updating using the Red Hat OpenShift OperatorHub

### Note:

Review [What's New](#) and [Fixed Issues](#) to see what is new for this release.

As Turbonomic evolves, the set of delivered probes change. When you update to a new version, the update does not change the properties of your Turbonomic instance or your probe configuration, as per what is defined in the Custom Resource. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

The steps to follow for updating your Turbonomic Platform depends on the Update approval strategy you specified when installing Turbonomic using the Red Hat OpenShift OperatorHub: Automatic or Manual. If you are unsure which approval strategy is in place, navigate to the **Operators > Installed Operators > Turbonomic Platform Operator > Subscription** tab. The **Update approval strategy** and **Upgrade status** are listed in the **Subscription details** section at the top of the screen.

### Note:

The **Starting version** listed on the **Subscription** tab indicates the Operator version, not the Turbonomic version. It is important to keep in mind that the Operator version does not drive the Turbonomic version. Although the Operator may be up to date, you must still manually update the Turbonomic server version for each update. See [Updates and Operator Version](#) for a listing of the product and Operator versions.

## Automatic Update approvals

If the Operator (or Turbonomic Platform Operator) Update approval strategy in the subscription is set to **Automatic**, the upgrade process initiates as soon as a new Operator version is available. In this case, you only need to update the Turbonomic version in the Custom Resource YAML.

### Note:

You cannot revert the version of either the installed Operator or Turbonomic Platform.

The Turbonomic Platform Operator, starting with v42.22, is automatically updated to pull from the IBM Container Registry (`icr.io/cpopen`). Ensure your operator is updated to the latest version, and the XL CR (`xl-release`) repository is set to the IBM Container Registry.

1. Select **Operators > Installed Operators > Turbonomic Platform Operator**.
2. Click the **Turbonomic Platform Operator** link under the **Provided APIs** header, then click **xl-release** under the **Name** header.
3. Click the **YAML** tab to open the Custom Resource YAML file.

(If not already done) Edit the Custom Resource YAML to change the `repository` parameter from the Red Hat Container Catalog to the IBM Container Registry (`icr.io/cpopen`).

Edit the Custom Resource YAML to change the version in the `tag: {version}` parameter to the product version to which you want to upgrade.

```
spec:
 global:
 customImageNames: false
```

```
repository: icr.io/cpopen/turbonomic
tag: {version}
```

**Note:**

The `customImageNames` parameter must be set to `false` when working with Red Hat Operators, and the `repository` parameter must specify `icr.io/cpopen/turbonomic` as the repository location.

4. Click **Save**.

5. Verify that the requested Turbonomic version installed correctly.

Open the **Workloads > Pods** page to confirm all pods have a **Status** listed as Running, and that the **Ready** column shows the expected pods are fully up, for example 1/1, 2/2, and so on. The upgrade process may take several minutes to complete.

6. If you are already logged into the Turbonomic UI, clear your browser data and refresh your browser, or open an incognito window.

You now have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

7. Notify other users who access the Turbonomic UI to clear their browser data and refresh their Turbonomic browser sessions.

## Manual Update approvals

If the Update approval strategy in the subscription is set to **Manual**, you must first manually approve pending upgrades for the Turbonomic Operator (`t8c-operator`), then update the Turbonomic server version in the Custom Resource YAML. Check [Updates and Operator Version](#) for a listing of the Turbonomic and Operator versions.

**Note:**

The Turbonomic Platform Operator, starting with v42.22, is automatically updated to pull from the IBM Container Registry (`icr.io/cpopen`). Ensure your operator is updated to the latest version, and the XL CR (`xl-release`) repository is set to the IBM Container Registry.

1. Select **Operators > Installed Operators**.

2. Operators that have a pending upgrade display a status with "Upgrade available." Click **Turbonomic Platform Operator**.

3. Select the **Subscription** tab. Any upgrades requiring approval are displayed next to **Upgrade status**. For example, it might display **1 requires approval**.

4. Click the **Upgrade status** link (for example, **1 requires approval**), then click **Preview Install Plan**.

5. Review the resources that are listed as available for upgrade. When satisfied, click **Approve**.

6. Navigate back to the **Operators > Installed Operators** page to monitor the progress of the upgrade. When complete, the status changes to "Succeeded" and "Up to date."

**Note:**

You may be presented with another operator update.

7. From the **Installed Operators** page, click **Turbonomic Platform Operator > Turbonomic Platform Operator** tab > **xl-release**.

8. Click the **YAML** tab to open the Custom Resource YAML file.

(If not already done) Edit the Custom Resource YAML to change the `repository` parameter from the Red Hat Container Catalog to the IBM Container Registry (`icr.io/cpopen`).

Edit the Custom Resource YAML to change the version in the `tag: {version}` parameter to the product version to which you want to upgrade.

```
spec:
 global:
```

```

customImageNames: false
repository: icr.io/cpopen/turbonomic
tag: {version}

```

**Note:**

The `customImageNames` parameter must be set to `false` when working with Red Hat Operators, and the `repository` parameter must specify `icr.io/cpopen/turbonomic` as the repository location.

9. Click **Save**.
10. Verify that the requested Turbonomic version installed correctly.  
 Open the **Workloads > Pods** page to confirm all pods have a **Status** listed as Running, and that the **Ready** column shows the expected pods are fully up, for example 1/1, 2/2, and so on. The upgrade process may take several minutes to complete.
11. If you are already logged into the Turbonomic UI, clear your browser data and refresh your browser, or open an incognito window.  
 You now have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle – usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.
12. Notify other users who access the Turbonomic UI to clear their browser data and refresh their Turbonomic browser sessions.

## Reference: What Are the Typical Settings for an IdP?

**Note:**

The process described here is applicable only to the OVA deployment model.  
 Before you begin configuring Single Sign-On (SSO), you need to make sure the IdP is set up for SSO.  
 Here are typical settings for a public Okta IdP which may be useful when you set up your IdP.

| SAML Settings: GENERAL                                                                                                                                                                   |                                                                                    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Setting                                                                                                                                                                                  | Example                                                                            |
| Single Sign On URL (where <b>&lt;hostname&gt;</b> is the host that Turbonomic runs on, and <b>&lt;samlRegistrationID&gt;</b> is the Registration ID that you got from your SSO provider) | <code>https://&lt;hostname&gt;/vmturbo/saml2/sso/&lt;samlRegistrationID&gt;</code> |
| Recipient URL (where <b>&lt;hostname&gt;</b> is the host that Turbonomic runs on, and <b>&lt;samlRegistrationID&gt;</b> is the Registration ID that you got from your SSO provider)      | <code>https://&lt;hostname&gt;/vmturbo/saml2/sso/&lt;samlRegistrationID&gt;</code> |
| Destination URL (where <b>&lt;hostname&gt;</b> is the host that Turbonomic runs on, and <b>&lt;samlRegistrationID&gt;</b> is the Registration ID that you got from your SSO provider)    | <code>https://&lt;hostname&gt;/vmturbo/saml2/sso/&lt;samlRegistrationID&gt;</code> |
| Audience Restriction                                                                                                                                                                     | <code>urn:test:turbo:markharm</code>                                               |
| Default Relay State                                                                                                                                                                      |                                                                                    |

**SAML Settings: GENERAL**

| Setting                                                                               | Example                                              |
|---------------------------------------------------------------------------------------|------------------------------------------------------|
| Name ID Format                                                                        | Unspecified                                          |
| Application username                                                                  | The username for the account that is managed by Okta |
| Response                                                                              | Signed                                               |
| Assertion Signature                                                                   | Signed                                               |
| Signature Algorithm                                                                   | RSA_SHA256                                           |
| Digital Algorithm                                                                     | SHA256                                               |
| Assertion Encryption                                                                  | Unencrypted                                          |
| SAML Single Logout                                                                    | Enabled                                              |
| Single Logout URL (where <b>&lt;hostname&gt;</b> is the host that Turbonomic runs on) | https://<hostname>/vmturbo/rest/logout               |
| SP Issuer                                                                             | turbo                                                |
| Signature Certificate                                                                 | Example.cer (CN=apollo)                              |
| authnContextClassRef                                                                  | PasswordProtectedTransport                           |
| Honor Force Authentication                                                            | Yes                                                  |
| SAML Issuer ID                                                                        | http://www.okta.com/\$(org.externalKey)              |

**SAML Settings: GROUP ATTRIBUTE STATEMENTS**

| Name  | Name Format | Filter                   |
|-------|-------------|--------------------------|
| group | Unspecified | Matches regex:.*admin.*. |

## Reference: FIPS Cipher Suites

**Note:**

The process described here is applicable only to the OVA deployment model.

The Federal Information Processing Standard (FIPS) is in place to ensure the cryptographic strength of secure connections. By default, Turbonomic ships with a FIPS-compliant cipher suite already enabled. The suite comprises the following ciphers:

- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384

- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384
- TLS\_ECDHE\_RSA\_WITH\_ARIA\_128\_GCM\_SHA256
- TLS\_ECDHE\_RSA\_WITH\_ARIA\_256\_GCM\_SHA384

## Modifying the Cipher Suite

If necessary, you can modify the cipher suite to comply with your internal policies.

1. Open an SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username:

```
turbo
```

- Password:

```
[your_private_password]
```

2. In the SSH session, open the cr.yaml file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

3. Edit the cipher suite.

Search for the list of ciphers in the file. Change the list as your policies require, and then save the file.

4. Apply your changes to the platform.

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

## Reference: Step-wise Platform Deployment

---

### Note:

The process described here is applicable only to the OVA deployment model.

After you have installed the Turbonomic VM that will host the platform, you can install the platform components, as follows:

1. Optionally, configure Single Sign-On Authentication (SSO) for this installation.

If you plan to use SSO to authenticate your Turbonomic users, you can configure it now. To configure SSO you will edit the `charts_v1alpha1_x1_cr.yaml` file. You can edit it now, before you complete the installation, or you can edit it later and restart the affected components. For more information, see [Single Sign-On Authentication](#).

2. Deploy Turbonomic Kubernetes nodes.

When you deploy Turbonomic on Kubernetes, you deploy one Kubernetes node as a VM that will host pods to run the Turbonomic components. The script to deploy and initialize the Kubernetes node also deploys the Kubernetes pods that make up the Turbonomic application.

Start a secure session (SSH) on your Turbonomic VM as the turbo user and perform the following steps:

- a. Initialize the Kubernetes node and deploy the pods.

Execute the script: `/opt/local/bin/t8c.sh`

Do not specify `sudo` when you execute this script.

The script should take up to 20 minutes to complete.

### Note:

You can specify the host name for the Turbonomic deployment.

By default, the Turbonomic platform deploys to a host named node1. To specify a different host name, execute the script with the `-h` flag as follows:

```
/opt/local/bin/t8c.sh -h <MyHostName>
```

Where `<MyHostName>`:

- Consists of lower-case alphanumeric characters, `!`, or `'`
- Begins and ends with an alphanumeric character

Also note that after you change the hostname, you must reboot the VM for the name change to take effect.

b. Verify that the deployment succeeded.

At the end of the script output, in the summary section, verify that no errors are reported. If any errors are reported, contact Turbonomic Support.

c. Verify that the Turbonomic application installed correctly.

```
kubectl get pods -n turbonomic
```

To verify the installation of the application, run the command:

```
kubectl get pods -n turbonomic
```

After all of the pods start, the `READY` column should read `1/1`, `2/2`, and so on and the `STATUS` column should read `Running` for each pod.

You should see output similar to the following:

| NAME                                       | READY | STATUS  | RESTARTS |
|--------------------------------------------|-------|---------|----------|
| action-orchestrator-b6454c9c8-mfl85        | 1/1   | Running | 0        |
| api-7887c66f4b-shndq                       | 1/1   | Running | 0        |
| arangodb-7f646fc5fc-zhcwf                  | 1/1   | Running | 0        |
| auth-5b86976bc8-vxwz4                      | 1/1   | Running | 0        |
| clustermgr-85548678d9-r5wb8                | 1/1   | Running | 0        |
| consul-7f684d8cb8-6r677                    | 1/1   | Running | 0        |
| cost-5f46dd66c4-6d6cb                      | 1/1   | Running | 0        |
| extractor-5f41dd61c4-4d6lq                 | 1/1   | Running | 0        |
| group-5bfd9bc6f8-96bsp                     | 1/1   | Running | 0        |
| history-5fc7fbc855-6zslq                   | 1/1   | Running | 0        |
| kafka-74cc77db94-dfrbl                     | 1/1   | Running | 0        |
| market-5f54699447-z4wkm                    | 1/1   | Running | 0        |
| mediation-actionscript-57b4fc6df-4lzfz     | 1/1   | Running | 0        |
| mediation-appdynamics-6d65f8766f-kb441     | 1/1   | Running | 0        |
| mediation-hpe3par-d7c475c4c-v8ftc          | 1/1   | Running | 0        |
| mediation-hyperv-6bd8c94df5-4dbzx          | 1/1   | Running | 0        |
| mediation-netapp-7f8fc955d9-4kkd1          | 1/1   | Running | 0        |
| mediation-oneview-7dbd7b54cf-7rfqp         | 1/1   | Running | 0        |
| mediation-pure-58c4bd8cd9-8n256            | 1/1   | Running | 0        |
| mediation-ucs-6f4bb9889-9rnqk              | 1/1   | Running | 0        |
| mediation-vcenter-5bc4f5fbd4-nzm4j         | 1/1   | Running | 0        |
| mediation-vcenterbrowsing-5c5987f66c-bfjq4 | 1/1   | Running | 0        |
| mediation-vmax-6c59969b89-28t9j            | 1/1   | Running | 0        |
| mediation-vmm-9c4878cf9-rfxnl              | 1/1   | Running | 0        |
| nginx-5b775f498-sm2mm                      | 1/1   | Running | 0        |
| plan-orchestrator-6dff4c9b6-p5t5n          | 1/1   | Running | 0        |
| reporting-b44fbd9b4-8fjv5                  | 1/1   | Running | 0        |
| repository-6d555bb4bf-fxldh                | 1/1   | Running | 0        |
| rsyslog-fd694878c-5tb2c                    | 1/1   | Running | 0        |
| t8c-operator-558bcc758d-5h8mp              | 1/1   | Running | 0        |
| topology-processor-b646b786b-9skp7         | 1/1   | Running | 0        |
| zookeeper-5f65b5bf69-nnmbt                 | 1/1   | Running | 0        |

d. Synchronize the system clock.

To ensure correct display of data, and to support Single Sign-On (SSO) authentication, you need to synchronize the system clock.

For information, see [Synchronizing Time](#) and [Single Sign-On Authentication](#).

e. Verify that the Load Balancer has installed correctly.

To verify the presence of the Load Balancer, execute the command:

```
kubectl get services -n turbonomic | grep LoadBalancer
```

You should see output similar to the following:

```
nginx LoadBalancer 10.10.10.10 10.10.10.11 443:32669/TCP,80:32716/TCP 17h
```

f. Configure mediation.

The installation script automatically enables a default set of mediation probes. After installation completes, you can change the set of enabled mediation probes (see [Enabling and Disabling Probe Components](#)).

For Turbonomic to manage your IT environment, it must attach to targets in your environment so it can perform discovery and execute actions. The combination of the processes of discovery and action execution is *mediation*. This release of Turbonomic supports mediation through the following targets. If you need to use additional targets that are not in this list, contact Turbonomic Support.

- Applications and Databases
  - Apache Tomcat 7.x, 8.x, and 8.5.x
  - AppDynamics 4.1+
  - AppInsights
  - Datadog
  - Dynatrace 1.1+
  - IBM WebSphere Application Server 8.5+
  - Instana release-242 or later
  - JBoss Application Server 6.3+
  - JVM 6.0+
  - SQL Server 2012, 2014, 2016, 2017, 2019, and 2022
  - MySQL 5.7.x
  - NewRelic
  - Oracle 11g R2, 12c, 18c, and 19c
  - Oracle WebLogic 12c
- Cloud Native
  - Kubernetes, including any compliant k8s distribution (Rancher, Tanzu, open source, etc.)
  - Cloud-hosted k8s services (AKS, EKS, GKE, IBM, Cisco IKS, ROKS, ROSA, etc.)
  - Red Hat OpenShift 3.11 and higher (OCP 4.x)
- Fabric and Network
  - Cisco UCS Manager 3.1+
  - HPE OneView 3.00.04
- Guest OS Processes
  - SNMP
  - WMI: Windows versions 8 / 8.1, 10, 2008 R2, 2012 / 2012 R2, 2016, 2019 and 7
- Hyperconverged
  - Cisco HyperFlex 3.5
  - Nutanix Community Edition
  - VMware vSAN



- Hypervisors
  - IBM PowerVM 8, 9, 10
  - Microsoft Hyper-V 2008 R2, Hyper-V 2012/2012 R2, Hyper-V 2016, Hyper-V 2019
  - VMware vCenter 7.0 and 8.0
- Orchestrator
  - ActionScript
  - Flexera One
  - ServiceNow
- Private Cloud
  - Microsoft System Center 2012/2012 R2 Virtual Machine Manager, System Center 2016 Virtual Machine Manager, and System Center Virtual Machine Manager 2019
- Public Cloud
  - Amazon AWS
  - Amazon AWS Billing
  - Google Cloud Platform (GCP)
  - GCP Billing
  - Microsoft Azure Service Principal
  - Azure Billing
  - Microsoft Enterprise Agreement
- Storage
  - EMC ScaleIO 2.x and 3.x
  - EMC VMAX using SMI-S 8.1+
  - EMC VPLEX Local Architecture with 1:1 mapping of virtual volumes and LUNs
  - EMC XtremIO XMS 4.0+
  - HPE 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
  - IBM FlashSystem running on Spectrum Virtualize 8.3.1.2 or later (8.4.2.0 or later recommended)
  - NetApp Cluster Mode using ONTAP 8.0+ (excluding AFF and SolidFire)
  - Pure Storage F-series and M-series arrays
- Virtual Desktop Infrastructure
  - VMware Horizon

For information about these targets, see [Target Configuration](#).

**Important:**

You should wait until all the platform components have started up, are running, and are fully ready before your first login. If you try to add a license or add a target to the platform before the components are all ready, the platform can fail to initialize correctly. After the components start up, in your Web browser, type the static IP address of your Turbonomic VM. Your browser redirects the login page for Turbonomic users.

3. Log in to the Turbonomic user interface and set the administrator user account password.

Turbonomic includes a default user account named `administrator` which has an ADMINISTRATOR role. As you log in for the first time, you must set your own password for that account. You can create or delete other accounts with the ADMINISTRATOR role, but your installation of Turbonomic must always have at least one account with that role.

In the login page, enter the information as required, and make a note of it.

- Use the default credential for **USERNAME**: administrator.
- Type a password for **PASSWORD**.  
The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.
- Type the password again to verify it for **REPEAT PASSWORD**.
- Click **Create Account**.

This is the account you will use to access the Turbonomic user interface with administrator permissions. Be sure to save the user interface administrator account credentials in a safe place.

**Note:**

The initial login is always for the default user account named administrator which has an ADMINISTRATOR role.

4. After you have logged in as administrator, you can create other user accounts, and you can give them various roles. For more information about user accounts and roles, see [Managing User Accounts](#).

**Note:**

For security reasons, you can create a different account with an ADMINISTRATOR role to serve as the main administrator of your Turbonomic installation, and then delete the default administrator account. But remember, you must always have at least one user account with administrator privileges.

## Reference: Step-wise Offline Update

---

**Note:**

The process described here is applicable only to the OVA deployment model.

Complete the following steps to perform a stepwise offline update of your Turbonomic installation.

1. Save a snapshot of your current Turbonomic VM.

Before updating, shut down (not power off) the Turbonomic VM.

```
sudo init 0
```

Then, perform a snapshot (or clone the VM). This provides a reliable restore point you can turn to in the event that trouble occurs during the update. After you have the snapshot, bring the VM back online.

2. Download and attach the ISO image to the VM that runs Turbonomic.

Refer to the email you received from IBM for links to the Turbonomic OVA file and to the ISO image.

3. Mount the ISO image by logging in to vCenter.

- a. In vCenter, navigate to the Turbonomic VM.
- b. Right-click the VM and choose **Edit Settings**.
- c. In the CD/DVD Drive drop-down menu, select **Datastore ISO**, then browse to the Turbonomic update ISO image and choose it.
- d. Ensure that **Connect at power on** is selected.

4. Log in to the Turbonomic VM.

Use SSH to log in to the Turbonomic VM using the turbo account and password.

5. If you are prompted to override the CD-ROM lock, select **Yes** and click **OK** to disconnect anyway and override the lock.

6. Mount the ISO image.

```
sudo mount /dev/cdrom /mnt/iso
```

7. Verify the correct version of the ISO image is mounted.

```
ls /mnt/iso
```

Verify that the ISO image contains the correct version for your update.

8. Load the latest Docker images.

```
sudo /mnt/iso/turboinstall.sh
```

This script loads all the images to the Turbonomic instance. If the load is successful, it displays a message similar to:

```
The t8c upgrade iso has been mounted
Image check:
=====

All images have been loaded

```

If the load does not succeed, the script lists any images that did not load, along with instructions to load them manually.

9. Run the following command as a single line command to update Turbonomic.

```
/mnt/iso/turboupgrade.sh | tee \
/opt/turbonomic/t8c_upgrade_$(date +%Y-%m-%d_%H_%M_%S).log
```

Wait until the script is finished.

10. Verify that you are running the correct version of MariaDB.

For this version of the product, Turbonomic supports MariaDB, version 10.5.20. Even after updating to this Turbonomic version, it is possible that your installation is running an earlier version of MariaDB.

While still in the SSH session, check the MariaDB version.

```
mysql -u root --password=my_pwd -e "SHOW VARIABLES LIKE 'version';"
```

The output should be similar to:

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| version | 10.5.20-MariaDB |
+-----+-----+
```

If the MariaDB version is earlier than 10.5.20, you must update your MariaDB. For complete instructions and information, see [Verifying your MariaDB Version](#).

11. Unmount the ISO image.

```
sudo umount /dev/cdrom
```

12. Clear your browser data and refresh your browser.

After clearing the browser data and refreshing your browser, you have full access to Turbonomic features. However, features that rely on current analysis data will not be available until after a full market cycle — usually 10 minutes. For example, the Pending Actions charts will not show any actions until after a full market cycle.

13. Optionally, enable new probes in your environment.

**Note:**

As Turbonomic evolves, the set of delivered probes change. Also, from one version to the next, the set of probes that are enabled by default can change. When you update to a new version, the update

does not change your probe configuration. An update to a newer version does not automatically enable any new probes in your deployment. If you want to take advantage of new probes in an update, then you must enable them manually.

For steps to enable new probes in your updated version, see [Enabling and Disabling Probe Components](#). Use these steps to edit the platform's `cr.yaml` file.

**Note:**

During the online or offline upgrade process, you should not apply these changes at this time.

14. Optionally, enable Turbonomic on Turbonomic.

*For updates from versions earlier than 7.22.4*

In Turbonomic, the supply chain can model your deployment as a Business Application. For updates from versions earlier than 7.22.4, you must enable the components that implement this feature.

For steps to enable these components in your updated version, see [Enabling and Disabling Probe Components](#). Use these steps to edit the platform's `cr.yaml` file, but *do not* apply those changes yet, via `kubectl`.

When you update the components in your `cr.yaml` file, specify the following:

```
kubeturbo:
 enabled: true
prometheus-mysql-exporter:
 enabled: true
mysql:
 user: root
 pass: vmturbo
prometheus:
 enabled: true
prometurbo:
 enabled: true
```

15. Verify that the Turbonomic application installed correctly.

```
kubectl get pods -n turbonomic
```

After all of the pods start, the `READY` column should read `1/1`, `2/2`, and so on and the `STATUS` column should read `Running` for each pod.

16. Notify other users to clear their browser data and refresh their Turbonomic browser sessions.

## Reference: Working with YAML Files

---

**YAML** is the primary file format to create and configure resources on kubernetes, including everything to do with the Turbonomic platform. The Custom Resource YAML provides a convenient, single place to define the majority of configuration details for Turbonomic. When editing YAML files, you must be careful to respect the file syntax, including indents in the file. General rules for edits include:

- Always uses spaces, not tabs, for all indentation. Since spacing and indentations matter, and can yield an invalid result or a parameter completely skipped over, you should work with YAML files with an editor that supports using vertical lines associated with indentations to visually spot a misalignment. If your editor of choice makes this difficult, you can use the `linux expand` utility when you're done, to convert tabs to equivalent spaces.
- Indention uses two spaces per level.
- Be careful to keep the same indentation for all properties in a given section.
- Never use the same property name twice in the same section. Doing this will render the YAML file invalid, though you will likely not see any notification of a problem. Rather, all but one of the property definitions will be silently ignored.

## Spacing matters

For the Turbonomic Custom Resource, indentation defines where parameters are applied (globally or to specific components), so ensure you line up the text appropriately. The following example shows examples of specifications that are applied at a global level (`{ "spec": { "global": { "tag": "8.6.4" } } }`) which sets the container image tag for all instances. Then indented we see properties that are global for the remote database (`{ "spec": { "properties": { "global": { "dbPort": "6033" } } } }`) describes a property of `dbPort` that would be set for remote DB connections.

Each line is indented two spaces from the higher level.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

## Combine properties correctly

A YAML file is read top down, and if there are different parameters that apply to the same component, they need to be combined. The following example shows a YAML where properties for the `ui` component of image tag *and* memory limit resources.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
 ui:
 resources:
 limits:
 memory: 4Gi
```

This YAML will not set both image tag and memory limit resources for the `ui` component because they are set in two different sections. The memory limits would be applied since that is the last section to be read, and it would overwrite the first set of properties with the image tag. The following YAML will set both image tag and memory limit resources for the `ui` component:

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 resources:
 limits:
 memory: 4Gi
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

To avoid having the same property name twice in the same section, search for an existing instance of the key you want to add. With the complex embedded structure in many k8s resources, this can be quite challenging; however, you can use an online validator that does not accept duplicate keys to check your entire CR file (such as [YAMLint - The YAML Validator](#)).

## Finding a property

In this documentation we refer to specific properties in the CR file using a "path" expression. For example, assume the path `/spec/global/repository` to designate a `repository` property in the file. You can find the property in the file as follows:

1. Find a line that says `spec :` with no indentation at all.
2. Between that line and the next unindented line (not counting comments, which start with `#`), find a line that says `global :` and is at the next level of indentation.
3. Between that line and the next line with the same indentation, find a line that starts with `repository :`. That line is where the addressed property is defined.

In this example, find the `repository` property specified by `/spec/global/repository`:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: X1
metadata:
 name: x1-release
spec:
 properties:
 global:
 repository: # This is NOT the correct property
 ...

 # Global settings
 global:
 repository: # This is the one we're after
```

The first `repository` property is not at `/spec/global/repository`, but at `/spec/properties/global/repository`.

## Chapter 4. SaaS Setup

Refer to this guide if you use a SaaS server to manage your cloud and on-prem workloads.

### Terminology

- SaaS Server

A SaaS server is an instance of Turbonomic that runs in a public cloud environment so that a large footprint is not required in your data center. This server can manage both cloud and on-prem targets, similar to a customer-hosted Turbonomic instance. Your Turbonomic representative provisions, maintains, and updates this server.

- SaaS Client (secure connector for hybrid support)

A SaaS client, also known as the secure connector for hybrid support, is a lightweight instance of Turbonomic that you install in your on-prem environment so that a SaaS server can manage on-prem targets (such as VMware vCenter). A SaaS client is not required if your SaaS server only manages cloud targets (AWS, Azure, or GCP) or cloud-based APM targets (such as Dynatrace).

The SaaS client requires an update each time your Turbonomic representative updates your SaaS server to a new version. The update process keeps the version of the SaaS client components (probes and tunnel) in sync with the server. If the SaaS client is unable to update for some reason, it will attempt to update periodically until it succeeds.

### Requirements

| SaaS Feature                       | Requirements                                                                                                                                                                                                                               | Supported Targets                                                                                                                            |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| SaaS server login                  | If you don't already have a SaaS server, talk to your Turbonomic representative.<br><br>Log in to a SaaS server using the server address, username, and password from your Turbonomic representative.                                      | N/A                                                                                                                                          |
| Management of cloud targets        | Contact your Turbonomic representative or <a href="#">submit an IBM Support ticket</a> to enable cloud targets.                                                                                                                            | <ul style="list-style-type: none"><li>• AWS</li><li>• Azure</li><li>• GCP</li></ul>                                                          |
| Management of APM targets          | Contact your Turbonomic representative or <a href="#">submit an IBM Support ticket</a> if you want to enable APM targets, and indicate which environment (cloud or on-prem) these targets are monitoring and where the targets are hosted. | <ul style="list-style-type: none"><li>• AppDynamics</li><li>• AppInsights</li><li>• Dynatrace</li><li>• Instana</li><li>• NewRelic</li></ul> |
| Management of cloud native targets | Contact your Turbonomic representative or <a href="#">submit an IBM Support ticket</a> to enable cloud-native targets.                                                                                                                     | <ul style="list-style-type: none"><li>• Azure Kubernetes Service (AKS)</li></ul>                                                             |

| SaaS Feature                         | Requirements                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Supported Targets                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | <ul style="list-style-type: none"> <li>• Amazon Elastic Kubernetes Service (EKS)</li> <li>• Google Kubernetes Engine (GKE)</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <p>Management of on-prem targets</p> | <p>Before adding on-prem targets to your SaaS server, perform the following tasks:</p> <ul style="list-style-type: none"> <li>• In your on-prem environment, install an instance using the Turbonomic OVA file.</li> </ul> <p>For detailed instructions, see <a href="#">Installing the vCenter Image for the SaaS client</a>.</p> <ul style="list-style-type: none"> <li>• Configure the SaaS client and then establish a secure connection with the SaaS server. If the SaaS client is behind a firewall, the firewall must allow access to a number of IP addresses and ports. Open the required ports supplied by your Turbonomic representative.</li> </ul> <p>After the SaaS client successfully establishes a connection with a SaaS server, you can begin to add your on-prem targets.</p> <p><b>Note:</b></p> <p>Do not use the SaaS client to enable cloud, cloud native, and cloud-based APM targets. Your Turbonomic representative should enable these targets for you.</p> <p>For detailed instructions, see <a href="#">Configuring the SaaS Client</a>.</p> | <p>The following is a list of supported targets for the SaaS client. Targets are disabled by default, unless otherwise noted.</p> <ul style="list-style-type: none"> <li>• Applications and Databases <ul style="list-style-type: none"> <li>– Apache Tomcat</li> <li>– Datadog</li> <li>– JBoss Application Server</li> <li>– JVM</li> <li>– SQL Server</li> <li>– MySQL</li> <li>– Oracle</li> <li>– Oracle WebLogic</li> </ul> </li> <li>• Fabric and Network <ul style="list-style-type: none"> <li>– Cisco UCS Manager– enabled by default</li> <li>– HPE OneView</li> </ul> </li> <li>• Guest OS Processes <ul style="list-style-type: none"> <li>– SNMP</li> <li>– WMI</li> </ul> </li> <li>• Hyperconverged <ul style="list-style-type: none"> <li>– Cisco HyperFlex</li> <li>– Nutanix Community Edition</li> </ul> </li> <li>• Hypervisors <ul style="list-style-type: none"> <li>– IBM PowerVM</li> <li>– Microsoft Hyper-V</li> <li>– VMware vCenter – enabled by default</li> </ul> </li> <li>• Orchestrator <ul style="list-style-type: none"> <li>– ActionScript</li> <li>– Flexera One</li> <li>– ServiceNow</li> </ul> </li> <li>• Private Cloud <ul style="list-style-type: none"> <li>– Virtual Machine Manager</li> </ul> </li> <li>• Storage <ul style="list-style-type: none"> <li>– EMC ScaleIO</li> </ul> </li> </ul> |



| SaaS Feature | Requirements | Supported Targets                                                                                                                                                                                                                                                                                                                                                      |
|--------------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |              | <ul style="list-style-type: none"> <li>- EMC VMAX</li> <li>- EMC VPLEX</li> <li>- EMC XtremIO</li> <li>- HPE 3PAR</li> <li>- IBM FlashSystem</li> <li>- NetApp Cluster Mode – enabled by default</li> <li>- Pure Storage – enabled by default</li> <li>• Virtual Desktop Infrastructure <ul style="list-style-type: none"> <li>- VMware Horizon</li> </ul> </li> </ul> |

### Limitations

Only one SaaS client should be connected to the SaaS server, except when additional SaaS clients are being used to manage Kubet turbo deployments. For more information see [Deploying Kubet turbo with the SaaS Client](#).

## Logging In to Turbonomic

To get started with the platform, open a web browser to your Turbonomic installation. The Turbonomic platform serves the user interface to your browser, where you can log in and get started managing your environment. In this way, you can access the unique capabilities of Turbonomic from any internet connection.

Before you can log in to your SaaS deployment, your enterprise must have a valid Turbonomic account. To get the hostname or URL of the SaaS instance of your Turbonomic account, contact your Turbonomic representative.

To log in to Turbonomic:

1. Navigate your Web browser to the Turbonomic installation.

Enter the SaaS server address that was provided by your Turbonomic representative.

2. Provide the user name and password for your account.

Your system administrator creates user accounts. Contact your system administrator for login information.

After you log in, the browser opens to the Home Page. This page is your starting point for sessions with the Turbonomic platform. From the Home Page you can see the overviews of your environment. You should bookmark this URL for future use.

## Installing the SaaS Client for OVA Deployments

A SaaS client, also known as the secure connector for hybrid support, is a lightweight instance of Turbonomic that you install in your on-prem environment so that a SaaS server can manage on-prem targets (such as VMware vCenter).

The first step to installing Turbonomic is to deploy the VM that will host the platform. You will install the image as a lightweight SaaS client.

After you install the vCenter image, you must configure the SaaS client and then establish a secure connection between it and the SaaS server. Once a connection is established, you can begin to add on-prem targets to your SaaS server.

## Installing the vCenter Image for the SaaS Client

The first step to installing Turbonomic is to deploy the VM that will host the platform. For vCenter environments, we deliver an OVA image for each bi-weekly release.

### Note:

The OVA image can be installed in your on-prem environment as:

- A standalone platform that manages both on-prem and cloud targets
- A lightweight SaaS client that enables a SaaS server in the public cloud to manage on-prem targets

This topic outlines the steps to install the image as a lightweight SaaS client. For instructions on installing the image as a standalone platform, see [OVA: Installing the vCenter Image for On-prem Environments](#).

## Minimum Requirements

The requirements for running a Turbonomic instance depend on the size of the environment you are managing. Turbonomic keeps a real-time representation of your environment in memory. The greater the number of entities to manage, and the more extensive the relationships between them, the more resources you need for the VM that runs Turbonomic. And as the VM requirements increase, so do the requirements for the physical machine that hosts the VM.

The requirements listed here are recommendations that you should keep in mind as you plan your Turbonomic deployment. After deploying, if you find that you need to change memory capacity, CPU capacity, or both for the VM, you can shut it down, make changes, and then power it up again to use the new capacity.

| Environment Size                  | Recommended Memory (GB) | Recommended vCPUs (cores) |
|-----------------------------------|-------------------------|---------------------------|
| Small<br>0-5k entities            | 32                      | 8                         |
| Medium<br>5k-80k entities         | 64                      | 16                        |
| Large<br>80k-200k entities        | 128                     | 32                        |
| Extra Large<br>200k-1.5M entities | 192                     | 48                        |

To install the Turbonomic OVA:

1. Download the Turbonomic installation package.

The email you received from IBM includes links to the Turbonomic download pages. You can get the installation package from there.

The installation package includes the `turbonomic_t8c-<version>-<XXXXXXXXXXXXXXXX>.ova` file where `<version>` is the Turbonomic version number and `<XXXXXXXXXXXXXXXX>` is the timestamp.

For example: `turbonomic-t8c-8.0.5-20190916164429000.ova`

The OVA file deploys as a VM with the Turbonomic components already installed.

2. Import the OVA file into your datacenter.

Use the vCenter Server client to import the OVA into your environment.

3. Deploy the Turbonomic VM.

Configure the VM that was deployed from the OVA file. Ensure that the physical machine hosting the VM meets the minimum requirements.

4. Open the remote console.

For the Turbonomic VM that you just deployed:

- a. Choose the **Summary** tab.
- b. Click **Launch Remote Console**.

5. Set up the Turbonomic System Administrator account.

- a. In the remote console, log in with the following default credentials:

- Username: turbo

*Do not* use the account name, root.

- Password: vmturbo

Then, you will be prompted to enter a new password.

- b. Enter your new password.

The new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

**Note:**

Be sure to save the changes account credentials in a safe place. For security reasons, this is the only account that can access and configure the Turbonomic VM.

- c. Enter your new password again to verify it.

6. Update the root password.

The platform uses the root account for certain processes, such as rolling up log messages in `/var/log/messages`. To ensure the account credentials are current, you must change the password:

- a. Open a SuperUser session.

- In the remote console, enter `su -`
- At the password prompt, enter the default password: vmturbo

- b. Reset a new password.

After you log in as root with the default password, the system prompts you for a New password. This new password must comply with the strong password policy (a mixture of upper- and lower-case letters, numbers, and a symbol). Only you will know this new password.

**Note:**

Be sure to save the root account credentials in a safe place.

- c. Exit the SuperUser session.

Enter `exit`.

7. Set up the static IP address.

Operation and management of the Turbonomic platform both require a static IP address for the VM. The installed VM includes the `ipsetup` script to perform this task. To run the script, execute the following:

```
sudo /opt/local/bin/ipsetup
```

When the script runs it requests the following inputs.

**Note:**

*You must provide values for these required fields.* Otherwise the installation can fail or your VM can be unreachable:

- **Required:** Do you want to use DHCP or set a static IP...  
Choose static
- **Required:** Please enter the IP Address for this machine
- **Required:** Please enter the network mask for this machine
- **Required:** Please enter the Gateway address for this machine
- **Required:** Enter DNS Server(s) IP Address for this machine

You should make a note of the IP address that you provide.

As part of running `ipsetup`, you can configure a proxy for internet access. To verify that your proxy has been properly set, first complete the full installation procedure. After you run the installation you can view the proxy configuration in `/etc/systemd/system/containerd.service.d/http-proxy.conf`. Find the `[Service]` section. It should contain your proxy and `no_proxy` settings. For example, it should be similar to:

```
Environment="HTTP_PROXY=http://10.10.12.34:3123" "NO_PROXY=<YourStaticIP>, node1, 127.0.0.1, 127.0.0.0"
```

## 8. Configure the SaaS client.

To review the prerequisites and perform the configuration steps for the SaaS client, see [Configuring the SaaS Client](#).

## Configuring the SaaS Client

After you have installed the vCenter image, you must configure the SaaS client and then establish a secure connection between it and the SaaS server. Once a connection is established, you can begin to add on-prem targets to your SaaS server.

### Note:

These steps are not required if your SaaS server only manages cloud and cloud-based APM targets.

## Prerequisites

### • Firewall Ports

If the SaaS client is behind a firewall, you must first ensure that the VM created from the OVA can make outbound requests to the following external IP address/port combinations. The `<tunnel-ip>` and `<tunnel-router-ip>` values are unique to your SaaS environment and will be provided by your Turbonomic representative.

| IP address                            | Port  | Direction | Description                                    |
|---------------------------------------|-------|-----------|------------------------------------------------|
| <code>&lt;tunnel-ip&gt;</code>        | 8081  | Outbound  | Used for initial token exchange                |
| <code>&lt;tunnel-router-ip&gt;</code> | 55671 | Outbound  | Used for sending data securely over the tunnel |
| <code>&lt;tunnel-router-ip&gt;</code> | 45671 | Outbound  | Used for sending data securely over the tunnel |

### Note:

Nothing needs to be changed on the VM created from the OVA. Any firewall changes are specific to your own environment.

## • Connection Verification

You can execute `/opt/local/bin/check_connection.sh` to verify that the required IP addresses and ports are accessible from the SaaS client environment.

For example, to verify the connection to `<tunnel-ip>` at port 8081, you would run the following command:

```
[turbo@node1 ~]$ /opt/local/bin/check_connection.sh --ip <tunnel-ip> --port 8081
```

You should see results similar to:

```
Checking connectivity to <tunnel-ip>:8081...
The request to <tunnel-ip>:8081 was successful
[turbo@node1 ~]$
```

## • Access to IBM Container Registry

For the self-upgrading probe capability to work, the client environment must be able to pull new images from the IBM Container Registry, if not using a private registry.

To allow the SaaS client to update automatically, add the following address to the allowlist:

```
https://icr.io
```

### Note:

Access to IBM Container Registry is mandatory to install the Turbonomic SaaS client and receive regular updates.

To verify the access to IBM Container Registry from the client environment, run the following command:

```
[turbo@node1 ~]$ nc -zv icr.io 443
```

## Installing the SaaS Client

Automatic updates are enabled by default to keep the SaaS client components (probes and tunnel) in sync with the server.

1. Open an SSH terminal session on your SaaS client instance.
2. Run the `t8cClientInstall.sh` script.

```
sudo /opt/local/bin/t8cClientInstall.sh
```

3. Press `y` if you have completed the network setup.

```
#####
 Initializing Kubernetes
#####
Have you run the ipsetup script to setup networking yet? [y/n] y
```

You should see the following result:

```
Turbonomic Client was installed successfully!
```

4. Verify the installation.

```
kubectl get pods
```

You should see results similar to the example below. Note that this abbreviated list is for illustration purposes only.

```
mediation-netapp-9f67f74cf-gplsk 0/1 Running 0 16m
mediation-pure-6ddc54f645-6s72b 1/1 Running 0 16m
```

**Note:**

Some probes (mediation-pods) may not become ready until the link has been established with the SaaS server.

**Establishing a Secure Connection Between the SaaS Client and the SaaS Server**

You must establish a secure connection between the SaaS client and the SaaS server. This requires creating a token and then applying the token to the SaaS client. If you have not been provided the URL or login credentials of your SaaS server, contact your Turbonomic representative before proceeding.

## 1. Create a new token.

a. Log in to your SaaS server.

b. Open the API page for your SaaS server using the following URL.

`https://mycompanyname.turbonomic.io/apidoc/#/Clients/createClientNetwork`

Replace `mycompanyname.turbonomic.io` with the domain name provided for the Turbonomic SaaS instance. For example:

`https://acme.turbonomic.io/...`

c. In the **Clients** section, find POST `/clients/networks/tokens`, click **Try it out**, and then click **Execute**.

The image shows two screenshots from the Turbonomic External API interface. The top screenshot displays the 'Clients' section with a list of endpoints. The endpoint `POST /clients/networks/tokens` is highlighted, and a 'Try it out' button is circled with a dashed box. An arrow points from this button to the bottom screenshot, which is a zoomed-in view of the 'Execute' button for the same endpoint. The 'Execute' button is also circled with a dashed box. The interface includes a 'Parameters' section with 'No parameters' and a 'Responses' section with a 'Response content type' dropdown set to 'application/json'.

d. Scroll to the **Response body** section and then click the Clipboard button to copy the entire token response.

The contents of the json token response should be similar to the following:

```
{
 "id": "5c3155c7-5a04-11ed-a12e-92430b25f882",
 "tokenData": "<base64 encoded token>"
}
```

## 2. Apply the token to the SaaS client.

- Open an SSH terminal session on your SaaS client instance.
- Execute the tokenExchange.sh script.

```
/opt/local/bin/tokenExchange.sh
```

- When prompted, paste the entire token that you obtained from the SaaS server.

The script should be similar to the following:

```
Enter token provided by your Turbonomic SaaS instance:
{
 "id": "5c3155c7-5a04-11ed-a12e-92430b25f882",
 "tokenData":
 "eyJhcGlWZXJzaW9uIjoidjEiLCJkYXRhIjpw7ImNhLmNydCI6IkkxTmHRMUzFDU1Vks1RpbQkRSVkpVU1VaS1EwR1VSU
zB0TFMwdENrMUpTVVJFYWt0RFFXWmhaMEYzU1VKQ1owbFNrVkJyVkdGblF1r0VRV2xTYUdWeFkzQkVSVVpKUTFGM1J
GR1pTa3R2V2t2sb2RtTk9RvKZGVEVKU1FY0tSMnBGV1UxQ1dVZEJNV1ZGUVhoT1VHTX1kREZqU0Vkc1kya3h1bUZZV
W14TVYwNW9UUVUkwVQVSVNyYbE5WRUY0VG5wRk1VMXFVWHPV05WVJGUkpNd3B0VkvGNFRtcEZNVTfxVVhwT1Zt0TN
SMnBGV1UxQ1dVZEJNV1ZGUVhoT1VHTX1kREZqU0Vkc1kya3h1bUZZVW14TVYwNW9UUVVxKUWtsCVFNUNAMnR4Q21oc
mFVYzVkekJDUVZGR1J1Rk1JUMESCVVROQ1RvBEPra05UzB0Q1VVVkJ0Vko0VjJoS116TTVWa1paV0U5YU9FVkhUSFK
0WVdsQmNgWm1UR1F5Vkv3S1EyRnNkV3hwWkVgDEsWkPNSGh2U1dwb04yMURPR1p2ZVcxVWJ1r11XRkKxw1RsV09Ha
3pURGRXWTFaSGNHdE1XSEo2WkdadWFr0TRiMUV3VkvZKd1p3cGtjUz1Hwkv4UmVXbHZaM1ZKVV5keF1UVm1VME42VkvZ
KcVRYVkpZa04wVG5kS1dWw1NRVnBESzIxU1RDc31RakU1TUZSa1RIaGx0a3h2WjJKYVZIR1ZDbWxYUW1zc116RTNUa
3BZTm5WcU9VNUdiM3BaT0dGd05YSjfiV2REU1c5UFdtnHdUbWhYUjBWSVHeFVabEZJw1VGNFJtNXhRM1ZLY250Tmf
sT1VhVtBLVTB0ak4xUk1VRVYzTDBSV1JHSmFMe1J5UTB3M1Uxw11UVzVsVmtWTVFqRnNSMFPvYkVoT2RveFJjVVRQT
ld3eU0wVXdUbFF6WwtJMFfXMXNhMDE0Wxdwd1JsQ1hUemM0WxPoeFZXDE5UbWhIV2t4N1RGSnVPSEpGvmt4W1ZtUjV
WbE13U1dWw1VXZG91RXB0TRkRwAFFqZzVMM2h2UkZGS1JFR1JRvUp2TURoM0NsU1VRVT1Dwja1V1NGRTRRa0ZtT0VWQ
1FVMURREZSZDB0UldVUldVakJzUwtKwMqWk1JXVWxMZDFsQ1Fsr1ZTRUYzU1VkrFEzTkhRVkZWUmtKM1RVtUtUVU
0UjBfEFZUXkZkMFZDTDNkU1J1MUJUvUpCwmpoM1EzZfPSR1pTTUzKQ1FWRjNRVz1KUVUxQk1FZERVM0ZIVTBSaU0WU
1JJSVUpEZDFWQ1FUUkpRZ3BCVVVKS1YxVTFSMVJEU1h0S1YzRmphVGRMv1hWbFprcE5PVWhQURGa1V1rNW5PVkppTTN
rMGNqR11RbEzrYjNkaU1ISk1iRwRaZDJKUGFVeEdhR3BoQ2pWdWnsVkhM0IyVkdsvmVgWnVLMk12U1ZaU0t6QTRhb
E5NwJJaUVozbHfHmWsxV25aS1NVRk11SHBrTDNsU1VbFVZVpMZG5sU1ZHSk9VvVd4UTA0S05uVnVjMmhoY1Rabk9
XNDRha1F5YUzKQmVtZHVXbVZ5V1RWR2FVZ3ZaREZfVEDESSWFXTJ0b1JsVG1vd2EwT1V1rVjXvmt0ek1GbHJNWEZpU
214M1JncG1RbFp3UWpoelNsSmthVFJwVUZVM1pISm9jVV13Y0UxN1JrS1FVMjv2T1RBck1rRjVjbbFF3UmpsR1JIV1R
ReXRMV1dkcU5VMWpTR3hqVhWc1VISm1Da0pJV1dGNVQysjB1SE5qTTJkV1JUbeTWWGhRYTFab1NFeG1LMUyYtkhwM
1JGSnVZV1pNVEdoewIyeE1kbVzhVDFkRVJubDVRM1YzTDNGVWVdC3h0ek1LUws5SmJFRnFURFI1YVRWTEszRmlRbGh
uWwXwaE1FOTfDaTB0TFMwdfJvNUVJRu5GVWxSS1JrbERRV1JGTFMwdeXTMESiLCJwYXNzd29yZCI6I1VXbHZ1bkJ1V
1UxwGvtVjN1RGd5TKU5WfPjXVjNwbT1MZVRST1ZXVjRTRk1UnpsdGVEBERVRXRETU5b1FVcGhTM2RqTmtSb1FqWmt
OMFZ3Y2toMwNYbE5WmWzWTJGefWf1MUxhbFZsYnpKT05Uwm1RbVv6YmXsTlpsS1dRMjVpYURoTGJqUjZaRFpEym1rM
mFFUTfHm001VDA5UFJuaFljRUZKY1RsQmNHdz0ifSwia2luZCI6I1N1Y3J1dCIsIm1ldGfKXhR1j7ImFubm90YXR
pb25zIjpw7InNrdXBwZXIuaW8vZ2VuzXJhdGvKlWJ5IjoInzAzMDc10WYtZGNhYy000DNkLW15MDYt0GR1ODE10DB1M
GN1Iiwic2t1cHBlc15pby9zaXRLLXZlcnNpb24iOiIxLjAuMSIsInNrdXBwZXIuaW8vZ2VuzXJhdGvKlWJ5IjoInzAzMDc10WYtZGNhYy000DNkLW15MDYt0GR1ODE10DB1M
yNDUuOTMumjEyojgwODEvNmMzMTU1YzctNWwNc0xMwVklWEXmMtOTIOMzBiMjVmODgyIn0sImxhYmVscyI6eyJza
3VvcGVyLm1vL3R5cGUioiJ0b2t1bi1jbGFPbS1J9LCluYw11IjoInWmMzMTU1YzctNWwNc0xMwVklWEXmMtOTIOMzBiMjVmODgyIn19"
```

You should then see the following results:

```
Token applied successfully. Waiting up to 30s for connection to be established...
Connection has been established.
Token exchange successful.
```

- Verify that the connection was established.

```
kubectl get services | grep ^remote
```

The results should be similar to:

|                     |           |               |        |                   |     |
|---------------------|-----------|---------------|--------|-------------------|-----|
| remote-nginx-tunnel | ClusterIP | 10.233.32.25  | <none> | 9080/TCP          | 96s |
| remote-rsyslog      | ClusterIP | 10.233.36.242 | <none> | 2514/TCP,8080/TCP | 96s |

These results indicate that the client-server connection has been established and the required client-side services are available. The probes for the on-prem targets should become ready.

- e. (Optional) It may take a few minutes for the probes to be ready. To speed up the process, manually delete the pods for the on-prem probes.

```
kubectl delete pod -l app.kubernetes.io/component=probe
```

New mediation pods should now appear and immediately connect to the remote services.

- f. Verify that all pods are now ready and running.

```
kubectl get pods
```

The on-prem probes are ready when you see results similar to:

| NAME                                                    | READY | STATUS  | RESTARTS | AGE |
|---------------------------------------------------------|-------|---------|----------|-----|
| mediation-netapp-66654c467c-f6711<br>3m47s              | 1/1   | Running | 0        |     |
| mediation-pure-6b86fbfbf7-2hmdv<br>3m47s                | 1/1   | Running | 0        |     |
| mediation-ucs-5c49577fc-k8ssc<br>3m42s                  | 1/1   | Running | 0        |     |
| mediation-vcenter-6fbdc997c7-87m8c<br>3m38s             | 1/1   | Running | 0        |     |
| mediation-vcenterbrowsing-557fd5f899-tmsjd<br>3m40s     | 1/1   | Running | 0        |     |
| skupper-router-7dbdf79c79-z8g98<br>3m37s                | 2/2   | Running | 0        |     |
| skupper-service-controller-76fcd64f6c-dcs1b<br>3m37s    | 1/1   | Running | 0        |     |
| skupper-site-controller-5f7fd46b-b5d1n<br>3m57s         | 1/1   | Running | 0        |     |
| t8c-client-operator-controller-manager-5c45fbb96f-6hc9h | 1/1   | Running | 0        | 77m |

**Note:**

You can execute the `/opt/local/bin/tunnelTroubleshooting.sh` script to inspect the deployment and report any errors it finds as well as advice on how to fix them.

## Adding On-prem Targets

Perform this task if you have the following on-prem targets. These targets are enabled by default, so you can add them to your SaaS server right away.

- Cisco UCS Manager
  - VMware vCenter
  - NetApp Cluster Mode
  - Pure Storage
1. Log in to your SaaS server.
  2. Navigate to **Settings > Target Configuration**.
  3. Click **New Target**, select the target type, and then select the target.
  4. Provide the required information and then add the target. See the Target Configuration topic for the target for additional information.

## Enabling Other On-prem Targets

By default, only a certain set of on-prem targets are enabled on the SaaS client. If you have any of the following targets, you should enable them first before adding them to your SaaS server.

**Important:**

Do not use the SaaS client to enable cloud, cloud native, and cloud-based APM targets. Your Turbonomic representative should enable these targets for you.

- Applications and Databases
  - Apache Tomcat



- JBoss Application Server
- JVM
- SQL Server
- MySQL
- Oracle
- Oracle WebLogic
- Fabric and Network
  - HPE OneView
- Guest OS Processes
  - SNMP
  - WMI
- Hyperconverged
  - Cisco HyperFlex
  - Nutanix Community Edition
- Hypervisors
  - IBM PowerVM
  - Microsoft Hyper-V
- Orchestrator
  - ActionScript
  - Flexera One
  - ServiceNow
- Private Cloud
  - Virtual Machine Manager
- Storage
  - EMC ScaleIO
  - EMC VMAX
  - EMC VPLEX
  - EMC XtremIO
  - HPE 3PAR
  - IBM FlashSystem
- Virtual Desktop Infrastructure
  - VMware Horizon

1. Open an SSH terminal session on your SaaS client instance.
2. If you have not done so, make a copy of the `turbonomicclient.yaml` file that shipped with the platform.

```
cp /opt/turbonomic/kubernetes/yaml/t8c-client-operator/turbonomicclient.yaml \
/opt/turbonomic/kubernetes/yaml/t8c-client-operator/turbonomicclient_backup.yaml
```

3. Edit `turbonomicclient.yaml` file to enable probes.

```
sudo vi /opt/turbonomic/kubernetes/yaml/t8c-client-operator/turbonomicclient.yaml
```

- Find the `spec:` section, followed by `global:`. Then, find a probe and then set the value for `enabled` to `true`.

```
spec:
 global:
 version: 8.7.3
 probes:
 actionScript:
 enabled: false
 appDynamics:
 enabled: true
 appInsights:
 enabled: true
```

**Note:**

To disable a probe, set the value for `enabled` to `false`.

- Apply the changes.

```
kubectl apply -f /opt/turbonomic/kubernetes/yaml/t8c-client-operator/turbonomicclient.yaml
```

- Log in to your SaaS server and then add the on-prem target you just enabled. See the Target Configuration topic for the target for additional information.

## Installing the SaaS Client using the Red Hat OpenShift OperatorHub

The Turbonomic SaaS client can be deployed in the Red Hat OpenShift Container Platform. This deployment brings all of the capabilities of the Turbonomic SaaS client to the Red Hat OpenShift Container Platform, such as secure communication to SaaS and auto-upgrade of both the probes and the operator.

This topic provides instructions for deploying the Turbonomic SaaS client in Red Hat OpenShift 4.8.x or higher using the Red Hat OpenShift OperatorHub.

### Prerequisites

- Firewall rules for Nginx and tunnel**

If the Red Hat OpenShift SaaS client is behind a firewall, you must first ensure that all nodes of the Red Hat OpenShift cluster can make outbound requests to the following external IP address/port combinations. The `<tunnel-ip>` and `<tunnel-router-ip>` values are unique to your SaaS environment and will be provided by your Turbonomic representative.

| IP address                            | Port  | Direction | Description                                    |
|---------------------------------------|-------|-----------|------------------------------------------------|
| <code>&lt;tunnel-ip&gt;</code>        | 8081  | Outbound  | Used for initial token exchange                |
| <code>&lt;tunnel-router-ip&gt;</code> | 55671 | Outbound  | Used for sending data securely over the tunnel |
| <code>&lt;tunnel-router-ip&gt;</code> | 45671 | Outbound  | Used for sending data securely over the tunnel |

- Access to IBM Container Registry**

For the self-upgrading probe capability to work, the client environment must be able to pull new images from the IBM Container Registry, if not using a private registry.

Add the following address to the allowlist for all nodes of the Red Hat OpenShift cluster:

```
https://icr.io
```

**Note:**

Access to IBM Container Registry is mandatory to install the Turbonomic SaaS client and receive regular updates.

• **Red Hat OpenShift cluster console access**

- Ensure you have log in access to the Red Hat OpenShift cluster console.
- Create a new project in the Red Hat OpenShift cluster.

This project is used throughout the installation process. All of the resources installed later, including the Turbonomic SaaS client, must be installed in this project (namespace).

1. Navigate to **Home > Projects**.
2. On the Projects page, click **Create Project**.
3. Enter the name of the project (for example, `turbonomic`) and click **Create**.

## Installing the SaaS Client Operator

1. Navigate to **Operators > OperatorHub**.
  2. In the **Filter by keyword** box, enter `Turbonomic Secure Connect Client` to find and select the Turbonomic SaaS client (Turbonomic Secure Connect Client).
  3. On the Turbonomic Secure Connect Client page, read the information about the Operator and click **Install**.
  4. On the Install Operator page, ensure the `Installed namespace` is the same as the project you created earlier during the prerequisites steps, for example `turbonomic`. Do not modify any other default configuration settings so that the operator configuration is as follows:
    - `Update channel`: `stable`
    - `Installation mode`: A specific namespace on the cluster
    - `Installed namespace`: `turbonomic`
    - `Update approval`: `Automatic`
- Note:**
- The `Update approval` must be set to `Automatic` to keep the SaaS client components (probes and tunnel) in sync with the server.
5. Click **Install**.
  6. Once the Operator installs and you see "Installed operator - ready for use," click **View Operator** or select **Operators > Installed Operators** to view the Turbonomic Secure Connect Client operator.
  7. Apply the Turbonomic Client custom resource.
    - a. On the Installed Operator page, click the **Details** tab for the Turbonomic Secure Connect Client operator.
    - b. On the Details tab, under **Provided APIs**, click **Create Instance** for the Turbonomic Client API.
    - c. On the Create TurbonomicClient page, select the **YAML** view to display the Custom Resource YAML file editor.
    - d. Edit the Custom Resource YAML to enable applicable targets for your instance. For a full description of supported targets, see [Target Configuration](#).
    - e. Ensure the selected **Project** is the same one in which you installed the Turbonomic Secure Connect Client operator.
    - f. Click **Create** to create an instance of the TurbonomicClient API.
  8. Apply the VersionManager custom resource.

- a. On the Installed Operator page, click the **Details** tab for Turbonomic Secure Connect Client operator.
- b. On the Details tab, under **Provided APIs**, click **Create Instance** for the Version Manager API.
- c. On the Create VersionManager page, ensure the selected **Project** is the same one in which you installed the Turbonomic Secure Connect Client operator.
- d. Click **Create** to create an instance of the VersionManager API.

## Establishing a Secure Connection Between the SaaS Client and the SaaS Server

You must establish a secure connection between the SaaS client and the SaaS server. This requires creating a token and then applying the token to the SaaS client. If you have not been provided the URL or login credentials of your SaaS server, contact your Turbonomic representative before proceeding.

### 1. Create a new token.

- a. Log in to your SaaS server.
- b. Open the API page for your SaaS server using the following URL.

`https://mycompanyname.turbonomic.io/apidoc/#/Clients/createClientNetwork`

Replace `mycompanyname.turbonomic.io` with the domain name provided for the Turbonomic SaaS instance. For example:

`https://acme.turbonomic.io/...`

- c. In the **Clients** section, find POST `/clients/networks/tokens`, click **Try it out**, and then click **Execute**.

The image shows two screenshots of the Turbonomic External API v3 interface. The top screenshot displays a list of API endpoints under the 'Clients' section. The endpoint 'POST /clients/networks/tokens' is highlighted, and a 'Try it out' button is circled. The bottom screenshot shows the expanded view for this endpoint, with the 'Execute' button circled. An arrow points from the 'Try it out' button in the top screenshot to the 'Execute' button in the bottom screenshot.

- d. Scroll to the **Response body** section and then click the Clipboard button to copy the entire token response.

The contents of the json token response should be similar to the following:

```
{
 "id": "5c3155c7-5a04-11ed-a12e-92430b25f882",
 "tokenData": "<base64 encoded token>"
}
```

## 2. Apply the token to the SaaS client.

After you create the token, the tokenData value of the token must be base64 decoded and applied as a Kubernetes Secret within the same project.

### a. Decode and apply the token to the SaaS client.

i) Decode the base64-encoded tokenData value from the response. Run the following command:

```
$ echo <base64 encoded token> | base64 -d
```

The JSON output should be similar to the following:

```
{
 "apiVersion": "v1",
 "data": {
 "ca.crt": "<base64 encoded certificate>",
 "password": "<base64 encoded password>"
 },
 "kind": "Secret",
 "metadata": {
 "annotations": {
 "skupper.io/generated-by": "aa7e96fe-bbcb-48b2-b2ce-a6a87b12a53d",
 "skupper.io/site-version": "1.2.4",
 "skupper.io/url": "<URL of SaaS instance used for token exchange>"
 },
 "labels": {
 "skupper.io/type": "token-claim"
 }
 },
 "name": "5c3155c7-5a04-11ed-a12e-92430b25f882"
}
```

ii) Copy the entire decoded Kubernetes Secret to the clipboard.

iii) On the Red Hat OpenShift UI, navigate to **Workloads > Secrets**.

iv) On the top-right side of the Secrets page, click **Create > From YAML**.

v) Replace the existing content with the copied JSON secret and click **Create**.

### b. Verify that the connection was established.

If the connection was successful, the remote-nginx-tunnel and remote-rsyslog services will be visible on the Services page.

i) Navigate to **Networking > Services**.

ii) In the **Filter by name** box, enter remote.

iii) Review to ensure the remote-nginx-tunnel and remote-rsyslog services are visible.

## Installing the SaaS Client on Other Deployments

The SaaS client can be installed on Kubernetes deployments other than the OVA and Red Hat OpenShift, such as Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS), and Google Kubernetes Engine (GKE). This topic provides instructions for deploying the SaaS client on other Kubernetes deployments using the Operator Lifecycle Manager (OLM).

### Installing the SaaS Client Operator

1. Open an SSH terminal session.

2. Create a namespace. The rest of the client installation steps are done in this namespace.

```
kubectl create namespace my-namespace
```

Switch to the new namespace.

```
kubectl config set-context --current --namespace my-namespace
```

3. Install the Operator Lifecycle Manager (OLM). The supported release version for OLM is v0.23.1.

```
curl -L https://github.com/operator-framework/operator-lifecycle-manager/releases/download/v0.23.1/install.sh -o install.sh
chmod +x install.sh
```

Run the `install` script to create all of the necessary k8s resources.

```
./install.sh v0.23.1
```

The results should be similar to:

```
customresourcedefinition.apiextensions.k8s.io/catalogsources.operators.coreos.com created
customresourcedefinition.apiextensions.k8s.io/clusterserviceversions.operators.coreos.com
created
customresourcedefinition.apiextensions.k8s.io/installplans.operators.coreos.com created
...
Package server phase: Installing
Package server phase: Succeeded
Waiting for deployment "packageserver" rollout to finish: 1 of 2 updated replicas are
available...
deployment "packageserver" successfully rolled out
```

Verify that the OLM installed correctly.

```
kubectl get csv -n olm
```

The results should be similar to:

| NAME          | DISPLAY        | VERSION | REPLACES | PHASE     |
|---------------|----------------|---------|----------|-----------|
| packageserver | Package Server | 0.23.1  |          | Succeeded |

4. Install the SaaS client operator.

To install the SaaS client operator, you must create yaml files for three entities: `CatalogSource`, `OperatorGroup`, and `Subscription`.

a. Create a new file called `catalogSource.yaml` and add the following content:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
 name: t8c-client-operator-catalog
 namespace: olm
spec:
 grpcPodConfig:
 securityContextConfig: restricted
 displayName: Turbonomic Client Operator
 image: icr.io/cpopen/t8c-client-operator-catalog
 publisher: Turbonomic.com
 sourceType: grpc
 updateStrategy:
 registryPoll:
 interval: 60m
```

Apply the file content.

```
kubectl apply -f catalogSource.yaml
```

b. Create a new file called `operatorGroup.yaml` and add the following content:

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
 name: turbonomic-operators
spec:
 targetNamespaces:
 - my-namespace
```

**Note:**

Ensure the namespace for `targetNamespaces` matches the current namespace.

Apply the file content.

```
kubectl apply -f operatorGroup.yaml
```

c. Create a new file called `subscription.yaml` and add the following content:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
 name: t8c-client-operator
spec:
 channel: stable
 name: t8c-client-operator
 source: t8c-client-operator-catalog
 sourceNamespace: olm
 installPlanApproval: Automatic
```

Apply the file content.

```
kubectl apply -f subscription.yaml
```

5. It may take a few minutes before the installation completes. Verify the installation by running the command:

```
kubectl get ClusterServiceVersion
```

The results should be similar to:

| NAME                       | DISPLAY             | VERSION | REPLACES                   |
|----------------------------|---------------------|---------|----------------------------|
| t8c-client-operator.v1.2.2 | t8c-client-operator | 1.2.2   | t8c-client-operator.v1.1.0 |
| Succeeded                  |                     |         |                            |

Verify that the `t8c-client-operator` pod logs do not have any errors.

```
kubectl logs deploy/t8c-client-operator-controller-manager
```

6. Create the SaaS client.

Create a new file called `turbonomicClient.yaml` and add the following content:

```
apiVersion: clients.turbonomic.ibm.com/v1alpha1
kind: TurbonomicClient
metadata:
 name: turbonomicclient-release
spec:
 global:
 version: 8.9.3
```

Apply the file content.

```
kubectl apply -f turbonomicClient.yaml
```

7. Create the Version Manager.

Create a new file called `versionManager.yaml` and add the following content:

```
apiVersion: clients.turbonomic.ibm.com/v1alpha1
kind: VersionManager
metadata:
 name: versionmanager-release
```

Apply the file content.

```
kubectl apply -f versionManager.yaml
```

## Establishing a Secure Connection Between the SaaS Client and the SaaS Server

You must establish a secure connection between the SaaS client and the SaaS server. This requires creating a token and then applying the token to the SaaS client. If you have not been provided the URL or login credentials of your SaaS server, contact your Turbonomic representative before proceeding.

1. Create a new token.

a. Log in to your SaaS server.

b. Open the API page for your SaaS server using the following URL.

`https://mycompanyname.turbonomic.io/apidoc/#/Clients/createClientNetwork`

Replace `mycompanyname.turbonomic.io` with the domain name provided for the Turbonomic SaaS instance. For example:

`https://acme.turbonomic.io/...`

c. In the **Clients** section, find `POST /clients/networks/tokens`, click **Try it out**, and then click **Execute**.

The image shows two screenshots of the Turbonomic External API interface. The top screenshot displays the 'Turbonomic External API v3' page with a sidebar menu containing 'Actions', 'Admin', 'Business Units', and 'Clients'. The 'Clients' section is expanded, showing three endpoints: 'GET /clients/networks/sites', 'GET /clients/networks/tokens', and 'POST /clients/networks/tokens'. A 'Try it out' button is highlighted with a dashed box. The bottom screenshot shows the 'POST /clients/networks/tokens' endpoint details, including a 'Parameters' section with 'No parameters', an 'Execute' button highlighted with a dashed box, and a 'Responses' section with a 'Response content type' dropdown set to 'application/json'. An arrow points from the 'Try it out' button in the top screenshot to the 'Execute' button in the bottom screenshot.



- d. Scroll to the **Response body** section and then click the Clipboard button to copy the entire token response.

The contents of the json token response should be similar to the following:

```
{
 "id": "5c3155c7-5a04-11ed-a12e-92430b25f882",
 "tokenData": "<base64 encoded token>"
}
```

2. Apply the token to the SaaS client.
  - a. Open an SSH terminal session on your SaaS client instance.
  - b. Install the jq package on your SaaS client instance. For details, see <https://jqlang.github.io/jq/download/>.
  - c. Run the following command in the same namespace that was created before installing the OLM:

```
$ jq -r .tokenData token.json | base64 -d | kubectl apply -f -
secret/ff108381-e034-11ed-86ed-dee462359331 created
```

## Verifying the Installation

The final step is to verify that the SaaS client and Version Manager installed correctly.

1. Verify that the SaaS client installed correctly.

```
kubectl get TurbonomicClient
```

The results should be similar to:

| NAME                     | VERSION |
|--------------------------|---------|
| turbonomicclient-release | 8.9.3   |

2. Verify that Version Manager installed correctly.

```
kubectl get VersionManager
```

The results should be similar to:

| NAME                   | PRODUCT | VERSION |
|------------------------|---------|---------|
| versionmanager-release | 8.9.3   |         |

## Deploying Kubeturbo with the SaaS Client

### Note:

The following steps are only applicable to clusters that have a SaaS client with an active connection to a SaaS server. Alternatively, Kubeturbo can be deployed without a SaaS client. For more information, see [Kubeturbo Deployment Options](#).

Kubeturbo uses the Turbonomic patented analysis engine to provide observability with control across the entire stack to assure the performance of running micro-services on Kubernetes platforms, and drives efficiency of underlying infrastructure. For more information, see [Kubeturbo documentation](#).

Kubeturbo can be deployed alongside the SaaS client to benefit from the following features:

1. Communication from Kubeturbo to the SaaS server uses a secure connection.
2. Kubeturbo automatically updates so that it is in sync with the SaaS server version.
3. Logs from Kubeturbo are sent to the SaaS server automatically.

## Installing Kubeturbo

Kubeturbo must be installed in the same Kubernetes namespace as the SaaS client and must be installed by using the Kubeturbo Operator. For more information, see [Kubeturbo Operator installation documentation](#).

For Red Hat OpenShift installations, see [Red Hat OpenShift Operator Hub documentation](#).

Kubeturbo with the SaaS Client is also supported on Azure Kubernetes Service (AKS), Amazon Elastic Kubernetes Service (EKS), and Google Kubernetes Engine (GKE) deployments. For more details, see [Installing the SaaS Client on Other Deployments](#).

## Configuring Kubeturbo

The SaaS client automatically detects if Kubeturbo is deployed in the same Kubernetes namespace and configures the Kubeturbo deployment.

### Note:

If the SaaS client is removed from the namespace, Kubeturbo must be manually reconfigured so that it no longer tries to use the Secure Tunnel. Update the `spec.serverMeta.turboServer` value in the Kubeturbo Custom Resource instance to point to the Turbonomic instance. For more information, see [Kubeturbo Operator installation documentation](#).

SaaS clients that are being used to manage Kubeturbo should not have other probes enabled.

---

## Chapter 5. Product Overview

Thank you for choosing Turbonomic, the premier solution for Application Resource Management (ARM) of cloud and virtual environments.

Application Resource Management is a top-down, application-driven approach that continuously analyzes applications' resource needs and generates fully automatable actions to ensure applications always get what they need to perform. It runs 24/7/365 and scales with the largest, most complex environments.

To perform Application Resource Management, Turbonomic represents your environment holistically as a *supply chain* of resource *buyers* and *sellers*, all working together to meet application demand. By empowering buyers (VMs, instances, containers, and services) with a budget to seek the resources that applications need to perform, and sellers to price their available resources (CPU, memory, storage, network) based on utilization in real-time, Turbonomic keeps your environment within the *desired state* — operating conditions that achieve the following conflicting goals at the same time:

- Assured application performance
  - Prevent bottlenecks, upsize containers/VMs, prioritize workload, and reduce storage latency.
- Efficient use of resources
  - Consolidate workloads to reduce infrastructure usage to the minimum, downsize containers, prevent sprawl, and use the most economical cloud offerings.

Turbonomic is a containerized, microservices architected application running in a Kubernetes environment (or within a VM) on your network or a public cloud VPC. You then assign services running on your network to be Turbonomic *targets*. Turbonomic discovers the entities (physical devices, virtual components and software components) that each target manages, and then performs analysis, anticipates risks to performance or efficiency, and recommends actions you can take to avoid problems before they occur.

---

### How Turbonomic Works

To keep your infrastructure in the desired state, Turbonomic performs Application Resource Management. This is an ongoing process that solves the problem of assuring application performance while simultaneously achieving the most efficient use of resources and respecting environment constraints to comply to business rules.

This is not a simple problem to solve. Application Resource Management has to consider many different resources and how they are used in relation to each other, and numerous control points for each resource. As you grow your infrastructure, the factors for each decision increase exponentially. On top of that, the environment is constantly changing — to stay in the desired state, you are constantly trying to hit a moving target.

To perform Application Resource Management, Turbonomic models the environment as a *market* made up of *buyers* and *sellers*. These buyers and sellers make up a *supply chain* that represents tiers of entities in your inventory. This supply chain represents the flow of resources from the datacenter, through the physical tiers of your environment, into the virtual tier and out to the cloud. By managing relationships between these buyers and sellers, Turbonomic provides closed-loop management of resources, from the datacenter, through to the application.

See [Supply Chain of Entities](#) for a visual layout of the buyer and seller relationships.

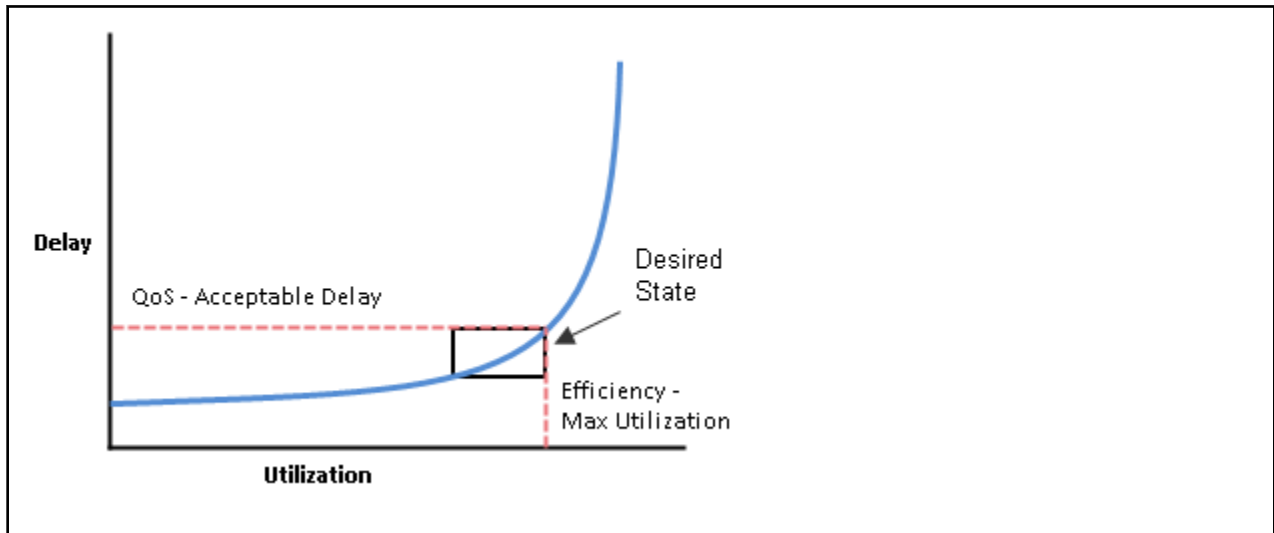
Turbonomic uses *Virtual Currency* to give a budget to buyers and assign cost to resources. This virtual currency assigns value across all tiers of your environment, making it possible to compare the cost of application transactions with the cost of space on a disk or physical space in a data center.

The price that a seller charges for a resource changes according to the seller's supply. As demand increases, prices increase. As prices change, buyers and sellers react. Buyers are free to look for other sellers that offer a better price, and sellers can duplicate themselves (open new storefronts) to meet

increasing demand. Turbonomic uses its *Economic Scheduling Engine* to analyze the market and make these decisions. The effect is an invisible hand that dynamically guides your IT infrastructure to the optimal use of resources.

To get the most out of Turbonomic, you should understand how it models your environment, the kind of analysis it performs, and the desired state it works to achieve.

## The Desired State



The goal of Application Resource Management is to assure performance while maintaining efficient use of resources. When performance and efficiency are both maintained, the environment is in the desired state. You can measure performance as a function of delay, where zero delay gives the ideal QoS for a given service. Efficient use of resources is a function of utilization where 100% utilization of a resource is the ideal for the most efficient utilization.

If you plot delay and utilization, the result is a curve that shows a correlation between utilization and delay. Up to a point, as you increase utilization, the increase in delay is slight. There comes a point on the curve where a slight increase in utilization results in an unacceptable increase in delay. On the other hand, there is a point in the curve where a reduction in utilization doesn't yield a meaningful increase in QoS. The desired state lies within these points on the curve.

You could set a threshold to post an alert whenever the upper limit is crossed. In that case, you would never react to a problem until delay has already become unacceptable. To avoid that late reaction you could set the threshold to post an alert before the upper limit is crossed. In that case, you guarantee QoS at the cost of over-provisioning — you increase operating costs and never achieve efficient utilization.

Instead of responding *after* a threshold is crossed, Turbonomic analyzes the operating conditions and constantly recommends actions to keep the entire environment within the desired state. If you execute these actions (or let Turbonomic execute them for you), the environment will maintain operating conditions that assure performance for your customers, while ensuring the lowest possible cost thanks to efficient utilization of your resources.

## The Market and Virtual Currency

To perform Application Resource Management, Turbonomic models the environment as a market, and uses market analysis to manage resource supply and demand. For example, bottlenecks form when local workload demand exceeds the local capacity — in other words, when demand exceeds supply. By modeling the environment as a market, Turbonomic can use economic solutions to efficiently redistribute the demand or increase the supply.

Turbonomic uses two sets of abstraction to model the environment:

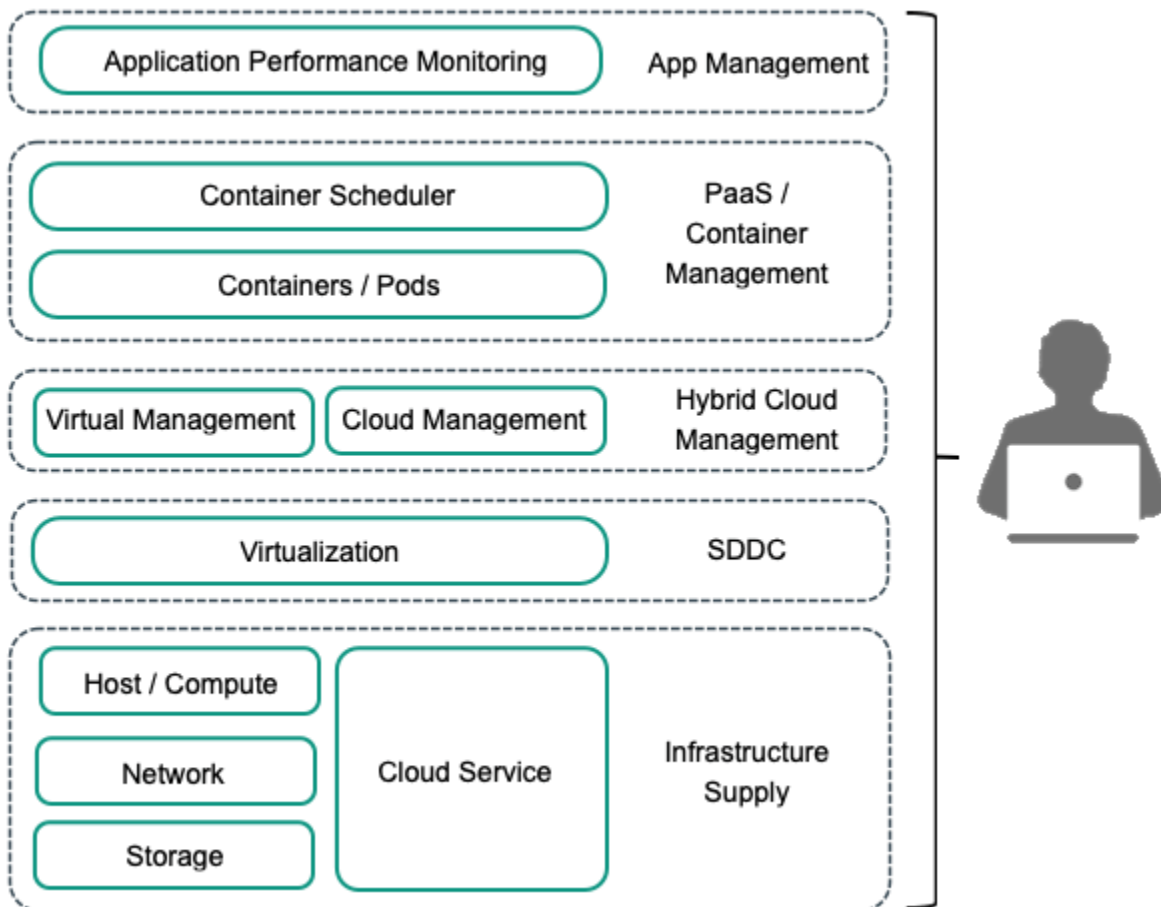
- Modeling the physical and virtual IT stack as a service supply chain

The supply chain models your environment as a set of managed entities. These include applications, VMs, hosts, storage, containers, availability zones (cloud), and data centers. Every entity is a buyer, a seller, or both. A host machine buys physical space, power, and cooling from a data center. The host sells resources such as CPU cycles and memory to VMs. In turn, VMs buy host services, and then sell their resources (VMem and VCPU) to containers, which then sell resources to applications.

See [Supply Chain of Entities](#) for a visual layout of the buyer and seller relationships.

- Using virtual currency to represent delay or QoS degradation, and to manage the supply and demand of services along the modeled supply chain

The system uses virtual currency to value these buy/sell transactions. Each managed entity has a running budget — the entity adds to its budget by providing resources to consumers, and the entity draws from its budget to pay for the resources it consumes. The price of a resource is driven by its utilization — the more demand for a resource, the higher its price.



These abstractions open the whole spectrum of the environment to a single mode of analysis — market analysis. Resources and services can be priced to reflect changes in supply and demand, and pricing can drive resource allocation decisions. For example, a bottleneck (excess demand over supply) results in rising prices for the given resource. Applications competing for the same resource can lower their costs by shifting their workloads to other resource suppliers. As a result, utilization for that resource evens out across the environment and the bottleneck is resolved.

## Risk Index

Turbonomic tracks prices for resources in terms of the *Risk Index*. The higher this index for a resource, the more heavily the resource is utilized, the greater the delay for consumers of that resource, and the greater the risk to your QoS. Turbonomic constantly works to keep the Risk Index within acceptable bounds.

You can think of Risk Index as the cost for a resource — Turbonomic works to keep the cost at a competitive level. This is not simply a matter of responding to threshold conditions. Turbonomic analyzes the full range of buyer/seller relationships, and each buyer constantly seeks out the most economical transaction that is available.

This last point is crucial to understanding Turbonomic. The virtual environment is dynamic, with constant changes to workload that correspond with the varying requests your customers make of your applications and services. By examining each buyer/seller relationship, Turbonomic arrives at the optimal workload distribution for the current state of the environment. In this way, it constantly drives your environment toward the desired state.

**Note:**

The default Turbonomic configuration is ready to use in many environments. However, you can fine-tune the configuration to address special services and resources in your environment. Turbonomic provides a full range of policies that you can set to control how the software manages specific groups of entities. Before you make such policy changes, you should understand default Turbonomic operation. For more information about policies, see [Working With Policies](#).

## The Turbonomic Supply Chain

Turbonomic models your environment as a market of buyers and sellers. It discovers different types of entities in your environment via the targets you have added, and then maps these entities to the supply chain to manage the workloads they support. For example, for a hypervisor target, Turbonomic discovers VMs, the hosts and datastores that provide resources to the VMs, and the applications that use VM resources. For a Kubernetes target, it discovers services, namespaces, containers, container pods, and nodes. The entities in your environment form a chain of supply and demand where some entities provide resources while others consume the supplied resources. Turbonomic *stitches* these entities together, for example, by connecting the discovered Kubernetes nodes with the discovered VMs in vCenter.

For information about specific members of the supply chain, see [Supply Chain of Entities](#).

### Supply Chain Terminology

IBM introduces specific terms to express IT resources and utilization in terms of supply and demand. These terms are largely intuitive, but you should understand how they relate to the issues and activities that are common for IT management.

| Term:       | Definition:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Commodity   | <p>The basic building block of Turbonomic supply and demand. All the resources that Turbonomic monitors are commodities. For example, the CPU capacity or memory that a host can provide are commodities. Turbonomic can also represent clusters and segments as commodities.</p> <p>When the user interface shows <i>commodities</i>, it's showing the resources a service provides. When the interface shows <i>commodities bought</i>, it's showing what that service consumes.</p>                                                                              |
| Composed Of | <p>The resources or commodities that make up the given service. For example, in the user interface you might see that a certain VM is <i>composed of</i> commodities such as one or more physical CPUs, an Ethernet interface, and physical memory.</p> <p>Contrast <i>Composed Of</i> with <i>Consumes</i>, where consumption refers to the commodities the VM has bought. Also contrast <i>Composed Of</i> with the commodities a service offers for sale. A host might include four CPUs in its composition, but it offers CPU Cycles as a single commodity.</p> |

| Term:       | Definition:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Consumes    | The services and commodities a service has bought. A service <i>consumes</i> other commodities. For example, a VM consumes the commodities offered by a host, and an application consumes commodities from one or more VMs. In the user interface you can explore the services that provide the commodities the current service consumes.                                                                                                                                                   |
| Entity      | A buyer or seller in the market. For example, a VM or a datastore is an entity.                                                                                                                                                                                                                                                                                                                                                                                                             |
| Environment | The totality of data center, network, host, storage, VM, and application resources that you are monitoring.                                                                                                                                                                                                                                                                                                                                                                                 |
| Inventory   | The list of all entities in your environment.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Risk Index  | <p>A measure of the risk to Quality of Service (QoS) that a consumer will experience. The higher the Risk Index on a provider, the more risk to QoS for any consumer of that provider's services.</p> <p>For example, a host provides resources to one or more VMs. The higher the Risk Index on the provider, the more likely that the VMs will experience QoS degradation.</p> <p>In most cases, for optimal operation the Risk Index on a provider should not go into double digits.</p> |

## Turbonomic Targets

**End-of-life (EOL) notice:** When a specific release or version of an integration partner technology reaches end-of-life (EOL) or its end of support date, Turbonomic no longer provides support for that version. Turbonomic follows integration partners' official EOL timeline for version support. Targeting a non-supported version, or one that is no longer supported by the vendor, is at your own risk.

You can assign instances of the following technologies as Turbonomic targets. Note that SaaS servers currently do not support WebSphere and vSAN.

- Applications and Databases
  - Apache Tomcat 7.x, 8.x, and 8.5.x
  - AppDynamics 4.1+
  - AppInsights
  - Datadog
  - Dynatrace 1.1+
  - IBM WebSphere Application Server 8.5+
  - Instana release-242 or later
  - JBoss Application Server 6.3+
  - JVM 6.0+
  - SQL Server 2012, 2014, 2016, 2017, 2019, and 2022
  - MySQL 5.7.x
  - NewRelic
  - Oracle 11g R2, 12c, 18c, and 19c
  - Oracle WebLogic 12c
- Cloud Native

- Kubernetes, including any compliant k8s distribution (Rancher, Tanzu, open source, etc.)
- Cloud-hosted k8s services (AKS, EKS, GKE, IBM, Cisco IKS, ROKS, ROSA, etc.)
- Red Hat OpenShift 3.11 and higher (OCP 4.x)
- Fabric and Network
  - Cisco UCS Manager 3.1+
  - HPE OneView 3.00.04
- Guest OS Processes
  - SNMP
  - WMI: Windows versions 8 / 8.1, 10, 2008 R2, 2012 / 2012 R2, 2016, 2019 and 7
- Hyperconverged
  - Cisco HyperFlex 3.5
  - Nutanix Community Edition
  - VMware vSAN
- Hypervisors
  - IBM PowerVM 8, 9, 10
  - Microsoft Hyper-V 2008 R2, Hyper-V 2012/2012 R2, Hyper-V 2016, Hyper-V 2019
  - VMware vCenter 7.0 and 8.0
- Orchestrator
  - ActionScript
  - Flexera One
  - ServiceNow
- Private Cloud
  - Microsoft System Center 2012/2012 R2 Virtual Machine Manager, System Center 2016 Virtual Machine Manager, and System Center Virtual Machine Manager 2019
- Public Cloud
  - Amazon AWS
  - Amazon AWS Billing
  - Google Cloud Platform (GCP)
  - GCP Billing
  - Microsoft Azure Service Principal
  - Azure Billing
  - Microsoft Enterprise Agreement
- Storage
  - EMC ScaleIO 2.x and 3.x
  - EMC VMAX using SMI-S 8.1+
  - EMC VPLEX Local Architecture with 1:1 mapping of virtual volumes and LUNs
  - EMC XtremIO XMS 4.0+
  - HPE 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
  - IBM FlashSystem running on Spectrum Virtualize 8.3.1.2 or later (8.4.2.0 or later recommended)
  - NetApp Cluster Mode using ONTAP 8.0+ (excluding AFF and SolidFire)
  - Pure Storage F-series and M-series arrays
- Virtual Desktop Infrastructure



## Resource Descriptions

To perform intelligent workload balancing, Turbonomic collects raw data from its target servers – hypervisors, cloud management stacks, public cloud accounts, etc. Turbonomic polls its targets at 10-minute intervals to collect the latest data samples. It then uses these 10-minute data points for analysis and to display data in the GUI.

The way Turbonomic collects host memory data from vCenter Server illustrates how this works. vCenter Server collects peak metrics from its managed VMs at 20-second intervals. Every ten minutes Turbonomic polls vCenter Server to collect its last round of data samples (30 samples in 10 minutes). To track a VM's utilization of host memory, Turbonomic requests *memory.active* data samples from vCenter. From that polling, Turbonomic can track:

- Peak Memory Utilization - Turbonomic uses the greatest value in each polling sample. This gives the highest percentage of active memory utilization for the selected VM (or group of VMs), calculated over the selected time period. For a maximum value, Turbonomic uses the highest observed active memory value in the data sample.
- Average Memory Utilization - Turbonomic averages all the values in each polling sample.

### Note:

The above example describes utilization calculations for on-prem entities. For workloads on the public cloud, Turbonomic includes the **Aggressiveness** and **Max Observation Period** settings to calculate a percentile of utilization. By using a percentile, Turbonomic can recommend more relevant actions to take advantage of elasticity on the public cloud.

The following table lists the metrics Turbonomic collects, and includes details about how they are collected or measured. When the Turbonomic user interface plots charts of clusters or groups of devices, these charts show the average of the percentage of allocated resources that are used.

| Resource:  | Description:                                                                                                                                                                                                                                                                                                                                                                                        |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Balloon    | <p>Ballooning capacity on the PM, measured in KBytes. This capacity is the greater of:</p> <ul style="list-style-type: none"> <li>• 65% of the VMem configured for all powered-on VMs that the PM hosts</li> <li>• The physical memory capacity of the PM</li> </ul> <p>Charts show the percentage of the PM's ballooning capacity that is in use.</p>                                              |
| Buffer     | <p>For network environments that support buffered switch ports (Arista networks), this resource measures utilization of a port buffer. For example, if a host connects to the network through port 1 on a switch, and that port has enough traffic to cause packet buffering, this resource will show utilization.</p>                                                                              |
| Connection | <p>Connection is the measurement of database connections utilized by applications.</p> <p>Turbonomic collects connection data from Database Servers discovered via Databases, APM, and Cloud targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the Connection chart.</p> <p>For details, see <a href="#">Connection Chart</a>.</p> |
| Cooling    | <p>Allocated cooling indicates the highest acceptable running temperature for a physical device, such as a chassis in a compute fabric.</p>                                                                                                                                                                                                                                                         |

| <b>Resource:</b>        | <b>Description:</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CPU                     | <p>Host CPU capacity, measured in MHz. This shows what percentage of CPU cycles are devoted to processing instructions.</p> <ul style="list-style-type: none"> <li>• Host charts show the percentage of the host’s CPU capacity that is in use.</li> <li>• VM charts show the percentage of the host’s CPU capacity that is consumed by the given VM.</li> </ul>                                                                                                                                                                      |
| DB Cache Hit Rate       | <p>DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.</p> <p>Turbonomic collects cache hit rate data from Database Servers discovered via Databases, APM, and Cloud targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the DB Cache Hit Rate chart.</p> <p>For details, see <a href="#">DB Cache Hit Rate Chart</a>.</p> |
| Database Memory (DBMem) | <p>Database memory (or DBMem) is the measurement of memory utilized by a Database Server.</p> <p>Turbonomic collects memory data from Database Servers discovered via Databases and APM targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the DB Memory chart.</p> <p>For details, see <a href="#">DB Memory Chart</a>.</p>                                                                                                                                          |
| Flow0 – InProvider Flow | <p>For measuring network flow, the flow that is within a single provider – For example, the network flow between VMs that are hosted by the same physical machine. This measures network flow between consumers that are on the same set of closely connected providers. Charts show the percentage of capacity that is utilized. Note that Turbonomic assumes an unlimited supply of InProvider Flow because this flow does not go across the physical network.</p>                                                                  |
| Flow1 – InDPOD Flow     | <p>For measuring network flow, the flow that is local to the given DPOD. This measures network flow between consumers that are on the same set of closely connected providers. Charts show the percentage of capacity that is utilized.</p>                                                                                                                                                                                                                                                                                           |
| Flow2 – CrossDPOD Flow  | <p>For measuring network flow, the flow that is between different DPODs. This measures network flow between consumers that are on different sets of closely connected providers. Charts show the percentage of capacity that is utilized.</p>                                                                                                                                                                                                                                                                                         |
| Heap                    | <p>Heap is the portion of a VM or container’s memory allocated to individual applications.</p> <p>Turbonomic collects heap data from Application Components discovered via Applications and APM targets. When you set the scope to one or several Application Components, the data that Turbonomic collected displays in the Heap chart.</p> <p>For details, see <a href="#">Heap Chart</a>.</p>                                                                                                                                      |
| HotStorage              | <p>For Nutanix platforms, the storage capacity on the server-attached flash.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| IO                      | <p>Data rate through the host’s IO adapter, measured in KBytes/sec.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Resource:                       | Description:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | <ul style="list-style-type: none"> <li>• Datacenter charts show the average percentage of the host IO capacity that is in use, for all the hosts in the datacenter.</li> <li>• Host charts show the percentage of the host's total IO capacity that is in use.</li> </ul>                                                                                                                                                                                                                                                                                          |
| IOPS                            | See 'Storage access'.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Latency                         | Allocated capacity for latency on a datastore. This measures the latency experienced by all VMs and hosts that access the datastore. Charts show the percentage of allocated latency that is in use on the datastore.                                                                                                                                                                                                                                                                                                                                              |
| Mem                             | <p>Host memory, measured in Kbytes.</p> <ul style="list-style-type: none"> <li>• Host charts show the percentage of the host's memory that is in use.</li> <li>• VM charts show the percentage of the host's memory that is consumed by the given VM.</li> </ul>                                                                                                                                                                                                                                                                                                   |
| NET                             | <p>Data rate through the host's Network adapter, measured in Kbytes/sec.</p> <ul style="list-style-type: none"> <li>• Datacenter charts show the average percentage of the host NET capacity that is used for all the hosts in the datacenter.</li> <li>• Host charts show the percentage of the host's total NET capacity that is in use.</li> </ul>                                                                                                                                                                                                              |
| Normalization factor (AWS only) | <p>Normalization factor is a measure of RI capacity that you can use to compare or combine the capacity for different instance families.</p> <p>Turbonomic measures RI coverage in terms of normalization factors. It compares the number of RIs calculated as normalization factors that cover workload capacity with the total number of normalization factors for a given Turbonomic scope. Each workload is assigned normalized units depending on its instance type.</p> <p>AWS normalization factor and Azure reservation ratio are equivalent concepts.</p> |
| Power                           | A measure of the power that is consumed by a physical device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Reservation ratio (Azure only)  | <p><u>Ratio</u> refers to the number of Azure reservation units that cover workload capacity compared to the total number of reservation units for a given Turbonomic scope. Each workload is assigned reservation units based on its instance type.</p> <p>Reservation ratio information appears in the tooltips of cloud discount charts. Information about the Azure instance types and their reservation workloads is provided in the Discount Inventory chart.</p> <p>Azure reservation ratio and AWS normalization factor are equivalent concepts.</p>       |
| Remaining GC Capacity           | <p>Remaining GC capacity is the measurement of Application Component uptime that is <i>not</i> spent on garbage collection (GC).</p> <p>Turbonomic collects GC data from Application Components discovered via Applications and APM targets, and then uses that data to calculate remaining GC capacity. When you set the scope to one or several Application Components, the capacity that Turbonomic calculated displays in the Remaining GC Capacity chart.</p> <p>For details, see <a href="#">Remaining GC Capacity Chart</a>.</p>                            |

| Resource:       | Description:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Response Time   | <p>Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).</p> <p>Turbonomic collects response time data from entities discovered via Applications, Databases, and APM targets. Entities include Business Applications, Business Transactions, Services, Application Components, and self-hosted Database Servers. When you set the scope to any of these entities, the data that Turbonomic collected displays in the Response Time chart.</p> <p>For details, see <a href="#">Response Time Chart</a>.</p> |
| Risk Index      | <p>A measure of the impact on Quality of Service (QoS) that a consumer will experience. The higher the Risk Index on a provider, the more risk to QoS for any consumer of that provider's services.</p> <p>For all the resources that impact performance or risk, charts show the Risk Index for the most utilized resource of a given entity. For example, if a host has a Risk Index of 6 for MEM and 12 for CPU, the chart will show the higher value.</p>                                                                                                                                                       |
| Storage access  | <p>Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.</p> <p>Turbonomic collects storage access data from VMs, Database Servers, and storage entities discovered via cloud, on-prem, and storage targets. When you set the scope to these entities, the data that Turbonomic collected displays in the Storage Access chart.</p> <p>For details, see <a href="#">Storage Access Chart</a>.</p>                                                                                                                                              |
| Storage amount  | <p>Storage Amount is the measurement of storage capacity utilized by the entity.</p> <p>Turbonomic collects storage amount data from VMs, Database Servers, and storage entities discovered via cloud, on-prem, and storage targets. When you set the scope to these entities, the data that Turbonomic collected displays in the Storage Amount chart.</p> <p>For details, see <a href="#">Storage Amount Chart</a>.</p>                                                                                                                                                                                           |
| Swap            | <p>The rate of memory swapping to disk, in bytes per second. The default capacity is 5,000,000 Byte/sec.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Threads         | <p>Threads is the measurement of thread capacity utilized by applications.</p> <p>Turbonomic collects thread data from Application Components discovered via Applications and APM targets. When you set the scope to one or several Application Components, the data that Turbonomic collected displays in the Threads chart.</p> <p>For details, see <a href="#">Threads Chart</a>.</p>                                                                                                                                                                                                                            |
| Transaction log | <p>Transaction log is the measurement of storage capacity utilized by Database Servers for transaction logging.</p> <p>Turbonomic collects transaction log data from Database Servers discovered via Databases and APM targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the Transaction Log chart.</p>                                                                                                                                                                                                                                            |

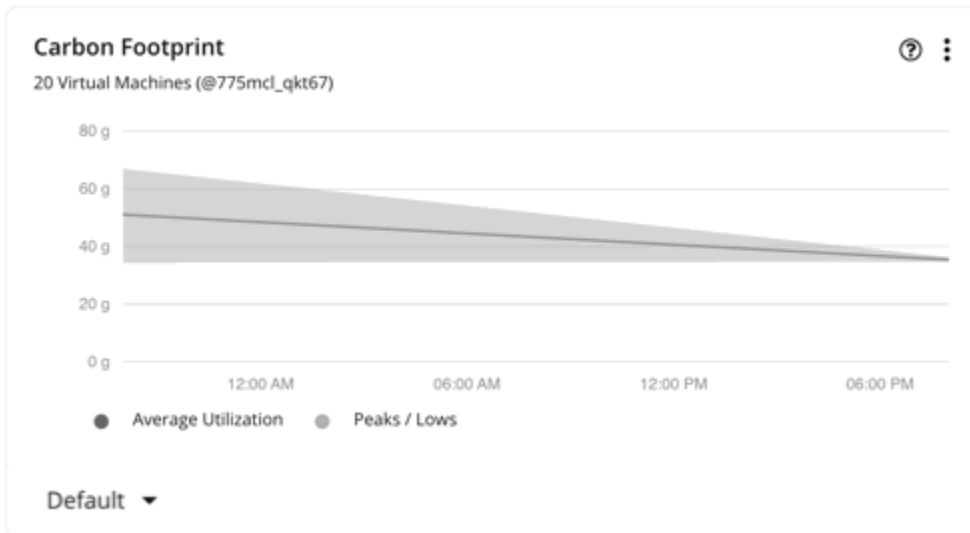
| Resource:       | Description:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | For details, see <a href="#">Transaction Log Chart</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Transactions    | <p>Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.</p> <p>Turbonomic collects transaction data from entities discovered via Applications, Databases, and APM targets. Entities include Business Applications, Business Transactions, Services, Application Components, and self-hosted Database Servers. When you set the scope to any of these entities, the data that Turbonomic collected displays in the Transaction chart.</p> <p>For details, see <a href="#">Transaction Chart</a>.</p> |
| VCPU            | The allocated CPU capacity, measured in MHz. Charts show the percentage of VCPU cycles that are devoted to processing instructions.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Virtual storage | <p>Virtual storage is the measurement of virtual storage capacity utilized by VMs.</p> <p>Turbonomic collects virtual storage data from VMs discovered via on-prem and APM targets. When you set the scope to one or several VMs, the data that Turbonomic collected displays in the Virtual Storage chart.</p> <p>For details, see <a href="#">Virtual Storage Chart</a>.</p>                                                                                                                                                                                    |
| VMem            | <p>The allocated memory capacity, measured in Kbytes. Charts show the percentage of VMem that is in use.</p> <p>Note that percentages of allocated VMem are measured against whichever is the less of: The VMem limit (if set) or the allocated VMem capacity. This is also true in reports and recommended actions. For example, assume a VM with allocated VMem of 8 GB, but a limit of 4 GB. In this case, the percentage in a chart shows the percentage utilized of 4GB.</p>                                                                                 |

## Sustainability Features

Turbonomic helps companies reduce their energy consumption and carbon footprint by optimizing IT infrastructure while ensuring that applications get the resources that they need to perform optimally. This core functionality, along with visibility into sustainability data, is crucial to setting and achieving sustainability goals. To visualize this data, Turbonomic provides the following features.

### Sustainability Charts

Turbonomic monitors the energy consumption and carbon footprint of your IT infrastructure, and then displays relevant data in charts. Currently, these capabilities are supported for hosts and VMs discovered via vCenter, UCS Manager and HPE OneView targets. When you set the scope to any of these entities and then click the **Details** tab, you can view sustainability data in the **Energy** and **Carbon Footprint** charts.



For details, see [Energy Chart](#) and [Carbon Footprint Chart](#).

## Data Center Policies

Turbonomic [calculates](#) carbon footprint using industry standards that take into account energy consumption, datacenter efficiency, and carbon intensity data. You can create Data Center policies to adjust the calculations according to the requirements of your data centers. For example, a data center in a particular location might have different requirements than data centers in other locations. After you adjust the calculations via policies, Turbonomic can accurately report your organization's carbon footprint.

**Configure Data Centers Policy**

NAME  
Datacenter Defaults

**OPERATIONAL CONSTRAINTS**

Carbon Intensity (g/Wh) 0.25 ⓘ

Power Usage Effectiveness 1.5 ⓘ

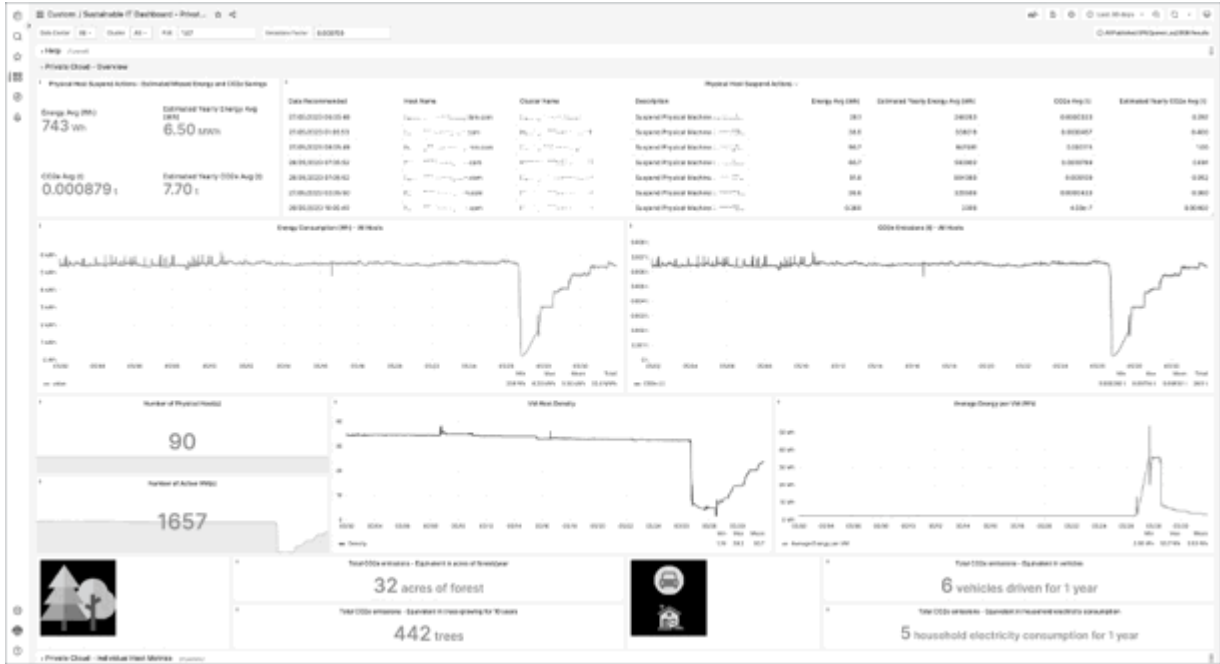
For details, see [Data Center Policies](#).

## Sustainability Dashboards in Embedded Reporting

If you have enabled the Embedded Reporting add-on, add the sustainability dashboards to gain insight into how you can reduce your carbon footprint and energy consumption.

- **Sustainable IT Dashboard – Private Cloud**

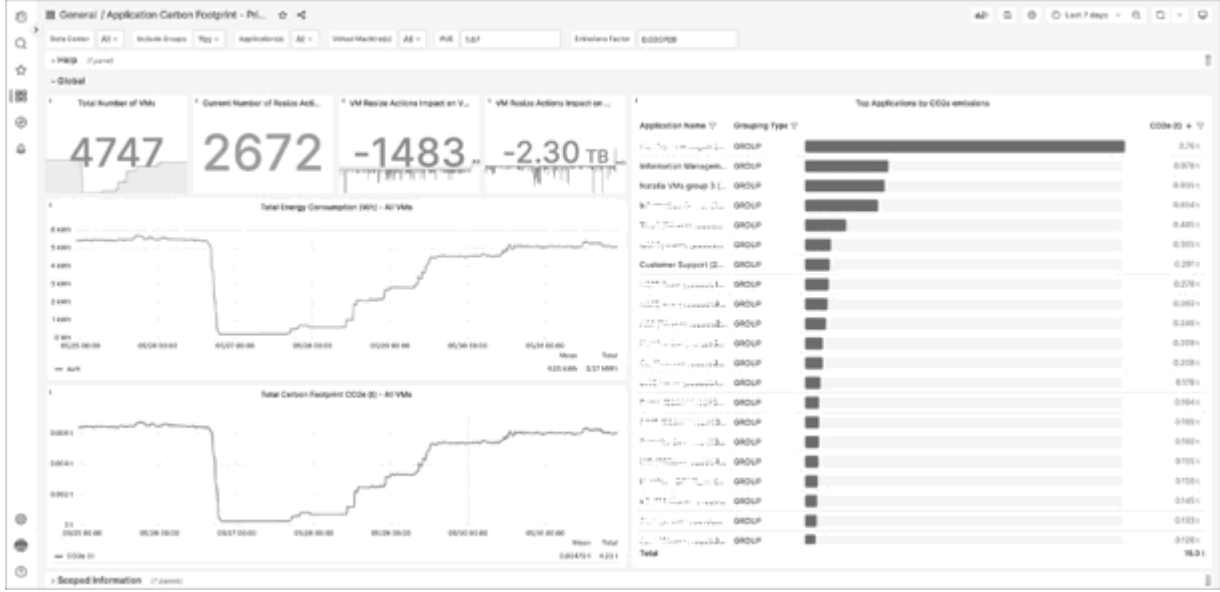
This dashboard focuses on the energy consumption and carbon footprint of hosts. It also shows opportunities to reduce consumption and footprint by executing Turbonomic actions.



You can change the scope of the dashboard by selecting data centers or clusters at the top left section.

• **Application Carbon Footprint – Private Cloud**

This dashboard focuses on the energy consumption and carbon footprint of VMs. The Top Applications by Carbon Footprint view provides insight into the carbon dioxide equivalent (CO2e) emissions of the VMs that power your applications. The dashboard also shows opportunities to reclaim resources by executing VM resize actions, as well as optimization opportunities that can help reduce consumption and footprint further.



You can change the scope of the dashboard by selecting data centers, applications, or VMs at the top left section. The expandable Help section explains the data and data collection methodology for some of the graphs.

For both dashboards, the expandable Help section explains the data and data collection methodology for some of the graphs.

To add these dashboards:

1. Enable Embedded Reporting. For details, see [Enabling Embedded Reporting](#).

2. Download the JSON files for the dashboards from [GitHub](#).
3. Log in to Turbonomic and then click **Reports** on the main menu.

The Embedded Reporting main page opens.

4. Click **Import** and then import the JSON files you downloaded. Be sure to select **Turbo Timescale** as your database.

The dashboards are now ready to use.



---

## Chapter 6. Getting Started

To get started with the platform, open a web browser to your Turbonomic installation. The Turbonomic platform serves the user interface to your browser, where you can log in and get started managing your environment. In this way, you can access the unique capabilities of Turbonomic from any internet connection.

### Logging In to Turbonomic

---

To get started with the platform, open a web browser to your Turbonomic installation. The Turbonomic platform serves the user interface to your browser, where you can log in and get started managing your environment. In this way, you can access the unique capabilities of Turbonomic from any internet connection.

Before you can log in, your enterprise must have a valid Turbonomic account, or an instance of Turbonomic must be installed in your environment. To get the IP address of your Turbonomic installation, contact your system administrator.

To log in to Turbonomic:

1. Navigate your Web browser to the Turbonomic installation.

For the URL, provide the IP address or machine name for the installation. This URL opens the Turbonomic Login page. You should bookmark this URL for future use.

2. Provide the user name and password for your account.

Your system administrator creates user accounts. Contact your system administrator for login information.

After you log in, the browser opens to the [Home Page](#). This page is your starting point for sessions with the Turbonomic platform. From the Home Page you can see the overviews of your environment.

To display this information, Turbonomic communicates with *target services* such as hypervisors, storage controllers, and public cloud accounts. Note that your Turbonomic administrator sets up the target configuration. For information about supported targets and how to configure them, see [Target Configuration](#).

## Product Walkthrough

### Note:

This topic discusses the product walkthrough experience for users who are interested in trying the *SaaS version* of the product.

If you wish to try the product, a 30-day trial is available from the Turbonomic website or AWS marketplace. When the product has been set up for you, log in to the user interface and review the following:

- The banner at the top-right section of the page

The banner indicates how long you have left in your trial. It also provides options to check pricing and initiate a purchase.

- Links to a product tour and video

Take a product tour or watch a video to learn more about Turbonomic and the many benefits that it offers.

- Getting started pages

These pages are intended to help you set up and use the product in three easy steps. These steps are described below.

## Connect Your Cloud Service

Connect your cloud service to see opportunities to optimize resources and improve overall performance. Turbonomic supports Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP).

Learn more about support for [AWS](#), [Azure](#), and [GCP](#).

## Review Your Dashboard

Within minutes of connecting a cloud service (usually 15 to 20 minutes), Turbonomic will start to build your dashboard with comprehensive visibility, insights, and recommendations.

Meanwhile, you can [visit our sandbox](#) to preview how Turbonomic can help you.


## Take Action

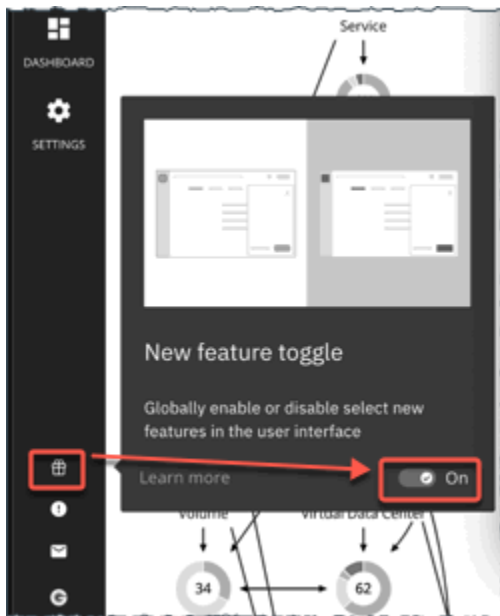
Turbonomic analyzes your environment in real time and then recommends actions to assure performance at the lowest possible cost. Use Action Center to review and execute these actions. You can automate actions to comply with service levels and improve business efficiency.

See a list of supported actions.

## Design Framework for the User Interface

We are transitioning the product to a new design framework to deliver a high-performance user interface with a more modern look-and-feel.

To switch to the new framework, click  on the main menu and then turn on the toggle.

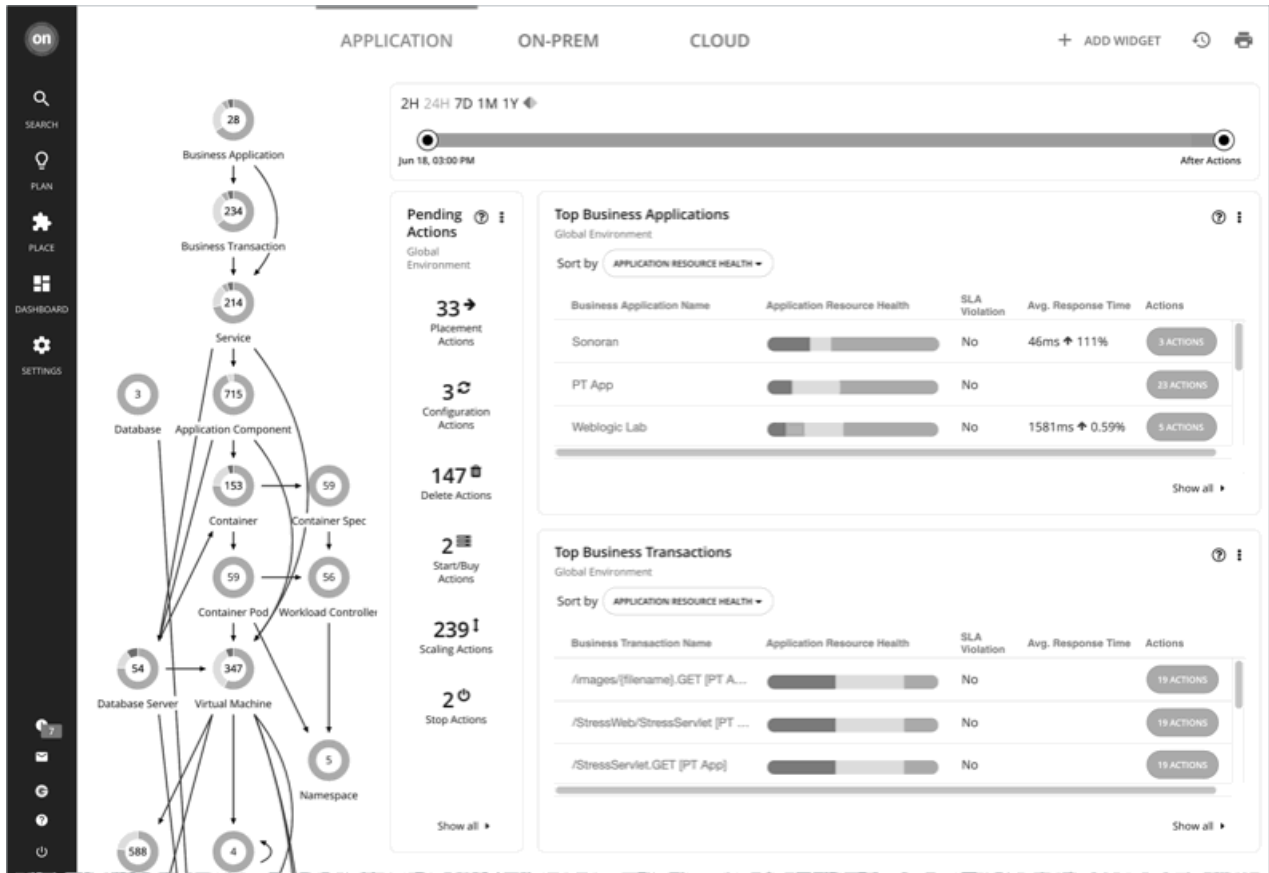


Currently, the following features and pages use the new framework:

- License Management
- Maintenance
- Updates
- Email Settings

We expect to complete the transition in the coming months. If you have feedback about the new framework, submit an Idea via the [IBM Ideas Portal](#).

# The Home Page



When you launch Turbonomic, the **Home Page** is the first view you see. From there you can:

- Choose a View to see overviews of your environment:
  - APPLICATION – See your environment in the context of your Business Applications.
  - ON-PREM – See details for the on-prem environment. Notice that the Supply Chain excludes cloud entities and only shows the entities that are on-prem.
  - CLOUD – See details for the cloud environment, including pending actions, a listing of your cloud accounts by cost, the locations of cloud datacenters that you are using, estimated costs, and other cost-related information.
- Use the Supply Chain Navigator to inspect lists of entities  
Click an entity tier in the Supply Chain to see a list of those entities. For example, click Virtual Machine to see a list of all the VMs in your environment.
- Navigate to other Turbonomic pages, including:
  - Search – Set the session scope to drill down to details about your environment
  - Plan – Run what-if scenarios
  - Place – Use Turbonomic to calculate the best placement for workloads, and execute the placement at the time you specify
  - Dashboards – Set up custom views with charts that focus on specifics in your environment
  - Settings – Configure Turbonomic to set up business rules and policies, configure targets, define groups, and perform other administrative tasks

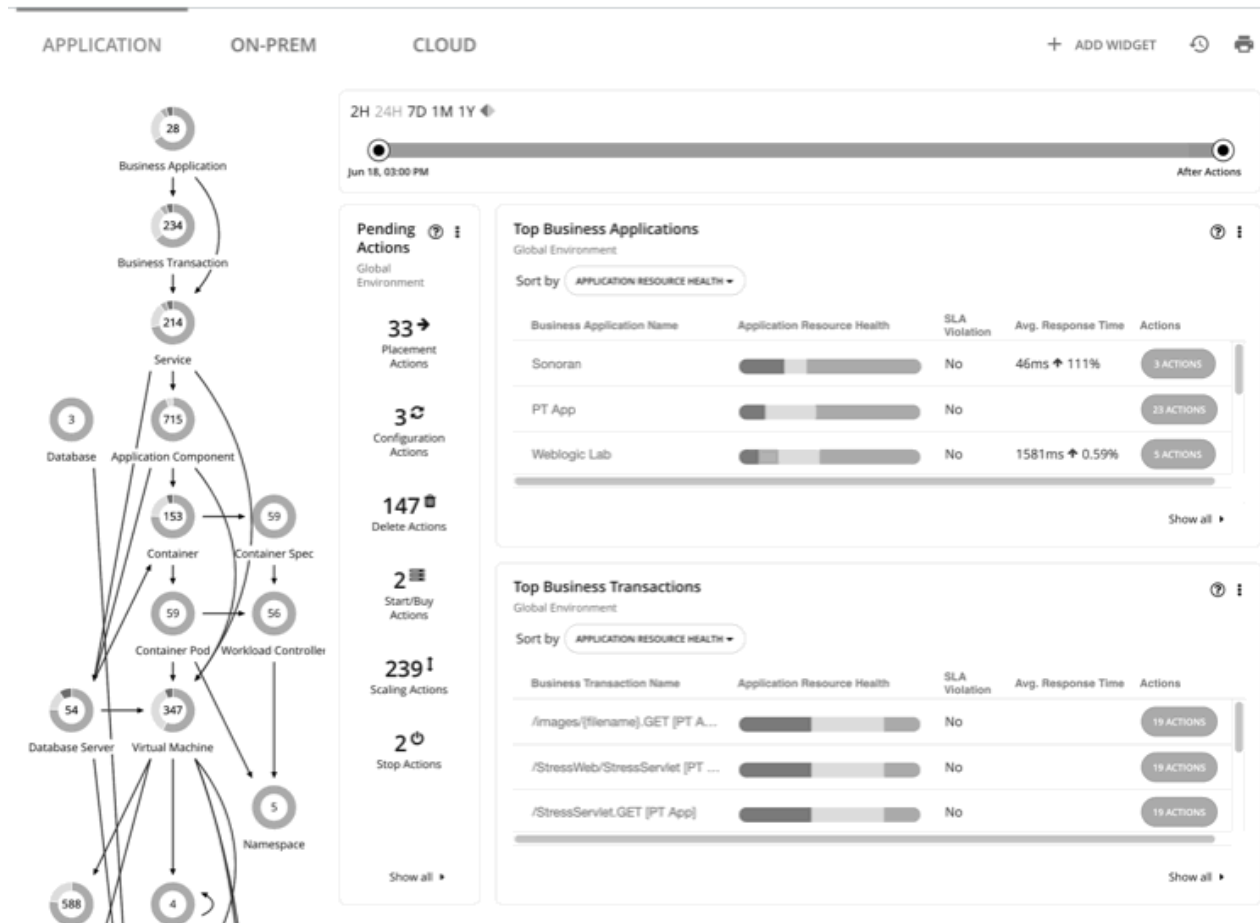
## Getting Home



Wherever you are in your Turbonomic session, you can always click the Home icon to return to the **Home Page**.

## APPLICATION View

The **APPLICATION** view presents your environment in the context of your Business Applications. See the overall health of your applications, examine any performance and compliance risks, and execute the actions that Turbonomic recommends to address these risks.



This view also shows the Business Transactions and Services that make up your Business Applications. You can see finer details and set SLOs at these levels of the application model.

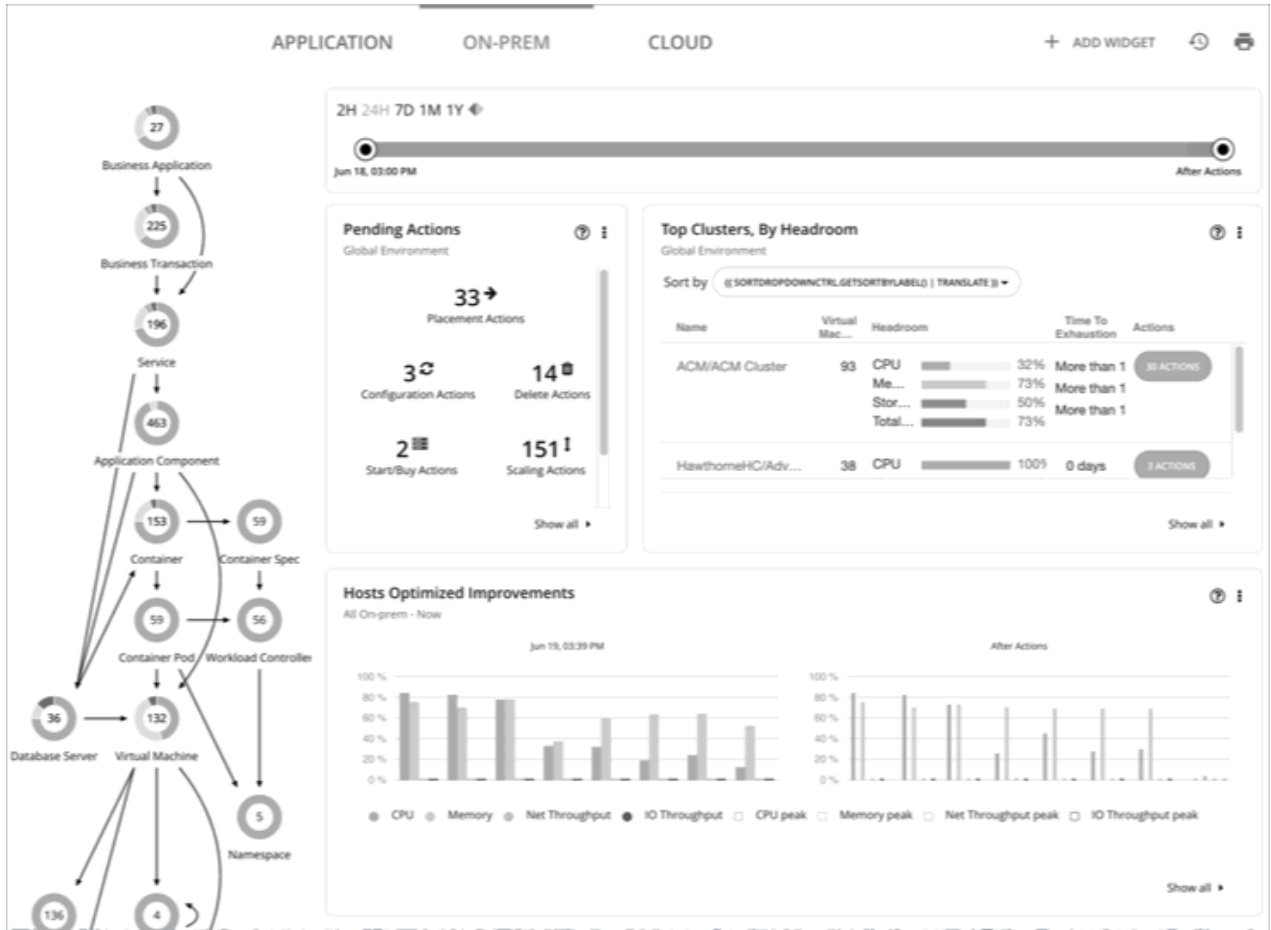
### Note:

If certain application entities do not stitch into the supply chain infrastructure for some reason, Turbonomic displays them in both the ON-PREM and the CLOUD views. Once Turbonomic can stitch them into the infrastructure, it classifies them according to the class of the infrastructure and displays them in the correct views.

## ON-PREM View

When you set your session to the Global Scope, you can then select the **ON-PREM** view. This shows an overview of your on-prem environment. If you don't have any workload on the public cloud, then

you should use this as your starting point for a Turbonomic session. If you have a hybrid environment (on-prem and on the public cloud), then you can refer to this view to see a detailed on-prem overview.



The Supply Chain shows all the on-prem entities in your environment. The charts show details about your environment, including:

- Overviews of pending actions

When appropriate, the overview includes estimated one-time savings or costs associated with the actions.

- Top Cluster utilization

See a list of the most utilized clusters. The chart shows these clusters, along with a count of actions for each. To drill down into the cluster details, click the cluster name. To see and execute the specific actions, click the **ACTIONS** button for that cluster. To see all the clusters in your environment, click **SHOW ALL**.

- Optimized Improvements

Compare current resource utilization with the utilization you would see if you choose to execute all the pending actions.

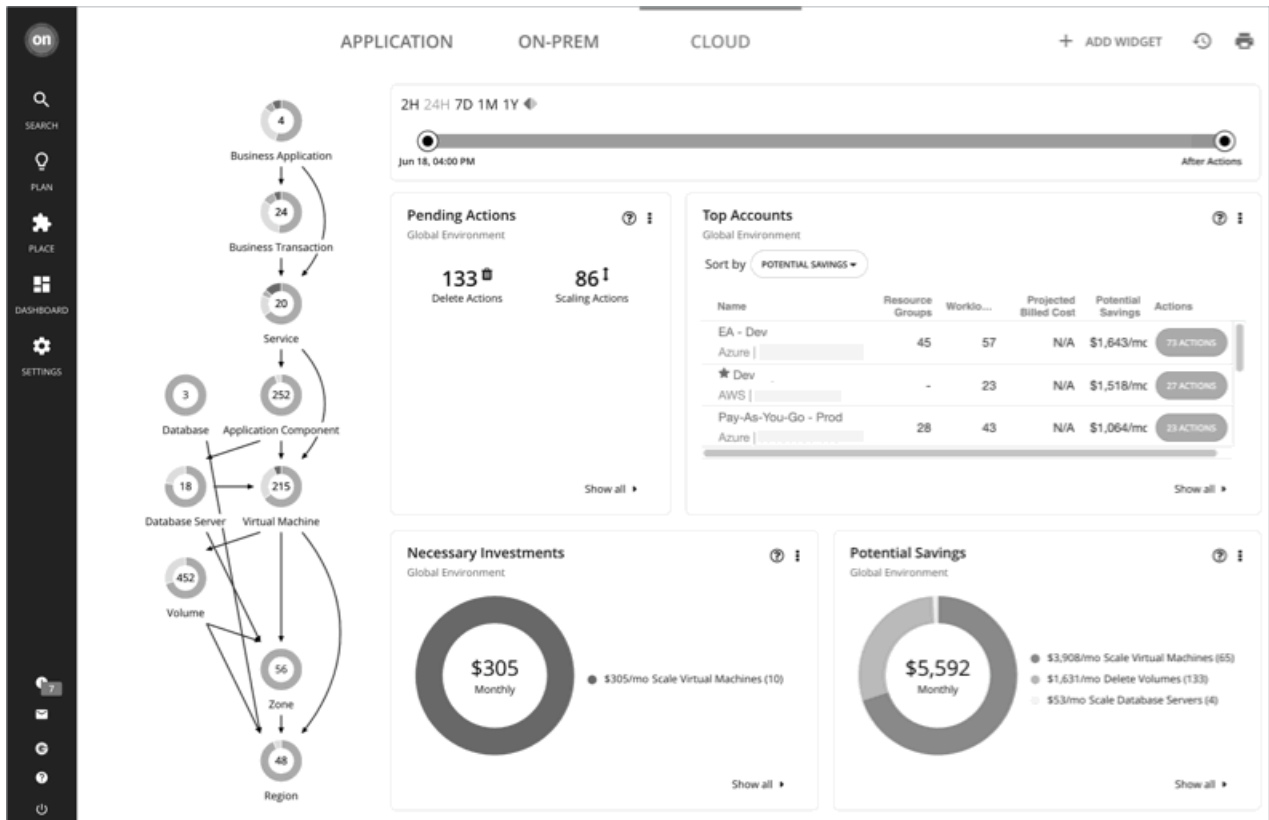
- Action history

You can see a history of all actions that have been recommended and executed, or of just the actions that have been accepted and executed.

## CLOUD View

When you set your session to the Global Scope, you can then select the **CLOUD** view. This shows an overview of your cloud environment. If all your workload is on the public cloud, then you should use this

as your starting point for a Turbonomic session. If you have a hybrid environment (on-prem and on the public cloud), then you can refer to this view to see a detailed cloud overview.



To view cloud cost information, you must have one or more public cloud targets set up in your Turbonomic installation. For information about setting up public cloud targets, see [Cloud Targets](#).

In addition, to view full cost information in AWS, you must have created a Cost and Usage report in your AWS account and you must store it in an S3 bucket. For details, see [Displaying AWS Spend In Turbonomic](#)

In this view, the Supply Chain shows all the cloud entities in your environment. The charts show details about your cloud environment, including:

- Overviews of pending actions

The overview includes the estimated monthly savings or cost associated with those actions.

- Top Accounts utilization

See a list of the most utilized public cloud accounts. The chart shows these accounts, along with an estimate of the monthly cost for each. To see all the cloud accounts in your environment, click **SHOW ALL**.

- Necessary Investments and Potential Savings

For the current set of pending actions, these charts show the impact in dollar value. Necessary Investments are from actions to provision more workloads or to resize workloads up. Potential Savings are from actions to resize down, or to purchase discounts and put them into active use.

- Charts that show your current discounts. For details, see [Discounts](#).
- Billed Cost by Service

This chart shows costs over time for each cloud service that you use in your cloud accounts. For example, you can see the cost for AWS CloudWatch, compared to the cost for AWS S3 storage.

## Tracking Cloud Cost

Turbonomic tracks your cloud costs based on the cost information it discovers from targets (for example, accounts, billing reports, and on-demand or discount costs) and [price adjustments](#).

### Cost for Services

Turbonomic uses the billing reports from your cloud service providers, as they are associated with your cloud targets. Turbonomic parses these reports to get cost breakdowns by service, service provider, Azure Resource Group, and cloud account. You can see cost data in the Expenses charts and Cost Breakdown by Tag charts.

### Workload Expenses

Workloads are the VMs running in your environment, or other hosted processes such as database servers and containers. Turbonomic tracks the following expenses for your workloads:

- Compute

For compute expenses Turbonomic uses hourly expense per template as specified in the associated public cloud account.

- Storage

Turbonomic discovers the storage tier that supports a given workload, and uses the tier pricing to calculate storage cost.

- License

For AWS environments, Turbonomic can calculate OS costs. To calculate the OS cost for a VM, Turbonomic subtracts the template cost from the published workload cost. It assumes the difference is the license cost for that workload. If the OS is open source, then there will be no difference, and license cost is zero. Analysis does not consider AWS Marketplace costs.

For Azure environments, Turbonomic can track OS costs for existing VMs. For actions to purchase reservations, Turbonomic does not include the OS cost. Analysis considers the base OS cost, but does not consider additional costs for support or other add-on features that are bundled with the OS. The affected OS types are Ubuntu PRO, SUSE 24/7, and RHEL with HA.

Turbonomic uses this cost information when making scaling decisions, both in real time and in plans. You can see this information in Expenses charts and in the results of Migrate to Cloud plans.

### Costs for Dedicated Tenancy on AWS

When you create VMs on AWS, you can specify their tenancy. When you specify Dedicated Tenancy (DT), the VMs you create are Amazon EC2 instances running on hardware that is dedicated to a single customer. To understand DT in the context of Turbonomic, you should consider:

- For AWS, the Turbonomic supply chain shows an Availability Zone as a Host. The supply chain does not indicate whether certain VMs have tenancy dedicated to specific resources in the given availability zone. Also, Turbonomic does not discover or show the costs for dedicated hosting of your workloads.
- Pricing for DT workloads is different than pricing for Shared Tenancy. Turbonomic does not discover that difference, and uses Shared Tenancy cost for the DT workloads. In action descriptions, the listed savings or investments will be based on Shared Tenancy costs.
- Turbonomic discovers the true costs of RIs for DT workloads. However, because the on-demand VM costs are based on Shared Tenancy, Turbonomic can overstate the savings you would get for purchasing and using RI capacity. In most cases, recommendations to purchase RIs will be correct. However, the time to achieve ROI could take longer than action descriptions and charts indicate.
- Some instance types that are valid for Shared Tenancy are not valid for DT. To see which instance types are valid for your DT VMs, consult the AWS documentation or your AWS representative.

- Under some circumstances Turbonomic can recommend changing a workload to a valid instance type for the tenant, even though the current type is already valid. This can happen when the instance type is not included in the Offer File for the tenancy. For example, assume the t3a template family does not support dedicated tenancy. However, assume that the user created a t3a instance with dedicated tenancy in the EC2 console. In that case, Turbonomic will see this as a misconfiguration and recommend changing to a different instance type.

To address these issues, you can create groups that set a scope to your DT workloads. For example, you can use naming conventions, tagging, or other means to identify your DT workloads. Then you can create dynamic groups based on those indicators. With those groups, you can create policies and dashboards that correspond to the differences you see in your DT environment. Use this approach to address issues for:

- Available Instance Types

To resize a workload, Turbonomic generates an action to change that workload to a different instance type. Because Turbonomic does not discover the difference between instance types that are valid for DT and for Shared Tenancy, it can recommend scaling a DT workload to an unavailable instance type. To avoid this, create a policy for the DT group, and exclude the unavailable instance types.

- Displaying Costs

Turbonomic charts show the costs for your environment. If the scope includes Dedicated Tenancy workloads, then the calculated cost will be incomplete. For example, since AWS does not return pricing data for converted RIs (that is, RIs that have been exchanged at least once) that are on *All Upfront* payment plans, Turbonomic does not include such RIs in its calculations of RI utilization or cost.

Use scope to minimize this effect. You can create separate dashboards for your DT and Shared Tenancy workloads.

## Discounts

Turbonomic analysis takes advantage of cloud provider discounts to calculate optimal workload placement and to arrive at the best possible costs for your deployments on the cloud. Turbonomic discovers the following discounts:

- AWS Reserved Instances (RIs) and Savings Plans
- Azure reservations
- GCP committed use discounts

The Cloud View in the Homepage includes the following charts that show discount data:

- [Potential Savings or Necessary Investments Charts](#)

If Turbonomic has found actions you can take to improve performance or to reduce cost, then you can see an overview of them in the Potential Savings or Necessary Investments charts. To see a listing of the specific actions, click **Show All** at the bottom of the chart. For more about actions, see [Turbonomic Actions](#).

- [Discount Utilization](#)

This chart shows how well you have utilized your current discount inventory. The desired goal is to maximize the utilization of your inventory and thus take full advantage of the discounted pricing offered by your cloud provider.

- [Discount Coverage](#)

This chart shows the percentage of VMs covered by discounts. If you have a high percentage of on-demand VMs, you should be able to reduce your monthly costs by increasing coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

- [Discount Inventory](#)

This chart lists the cloud provider discounts discovered in your environment.

- [Recommended RI Purchases](#)



Turbonomic can recommend purchasing instance types at a discounted rate to help you increase the percentage of VMs covered by discounted pricing and reduce on-demand costs. This chart shows your pending purchases. Download the list of purchases and then send it your cloud provider or representative to initiate the purchase process.

**Note:**

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

Currently, Turbonomic can recommend purchase actions for AWS and Azure. Purchase actions for GCP will be introduced in a future release.

## Configuring Targets

---

A target is a service that performs management in your virtual environment. Turbonomic uses targets to monitor workload and to execute actions in your environment. When you configure a target, you specify the address of the service, and the credentials to connect as a client to it.

For each target, Turbonomic communicates with the service via the management protocol that it exposes — The REST API, SMI-S, XML, or some other management transport. Turbonomic uses this communication to discover the managed entities, monitor resource utilization, and execute actions.

To configure a target, you will choose the target type, specify the target's address or key, and then provide credentials to access the target. Turbonomic then discovers and validates the target, and then updates the supply chain with the entities that the target manages.

**Note:**

Turbonomic regularly checks the status of your targets. If target discovery or validation fails, the Target Configuration page updates the status. Under some circumstances, the target can become discoverable or valid again, but the status does not update. In this case, select the target and then click **Rediscover** or **Validate**.

You have the option of using the enhanced target settings experience using the new target view. See [Configuring Targets - Enhanced View](#).

For a list of supported targets and configuration requirements, see [Target Configuration](#).

### Configuring a Target

1. Navigate to the Settings Page.



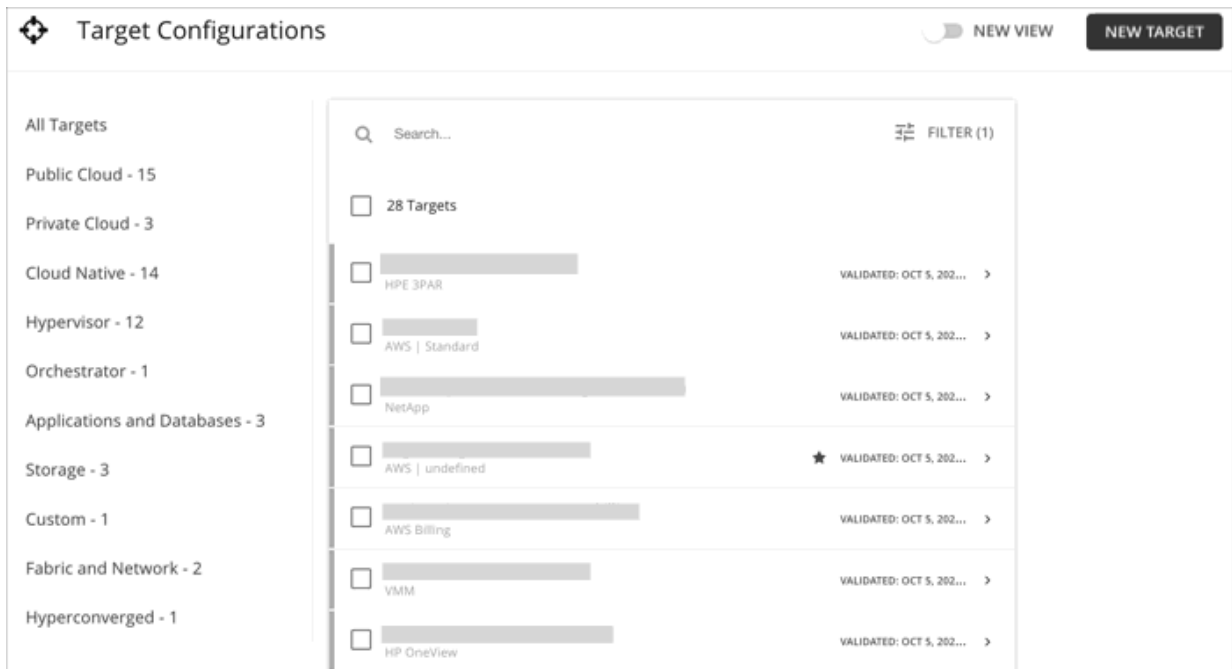
Click to navigate to the Settings Page. From there, you can perform a variety of Turbonomic configuration tasks.

2. Choose Target Configuration.



Click to navigate to the Target Configuration Page.

3. Review the list of targets.



This page lists all the targets that you currently have configured for Turbonomic. Click New View to enable the enhanced target settings experience using the new target view. You can inspect or edit these targets, or add a new target.

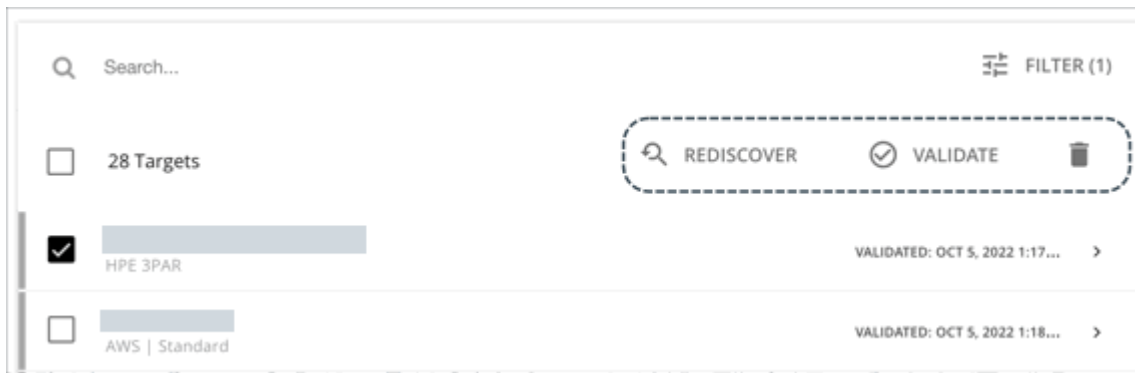
4. Filter the list of targets.



For a long list of targets, you can:

- Filter targets by target type.
- Use Search to filter targets by text string (partial matching is supported).
- Use Filter to filter targets by status (for example, only show validated targets). You can also use Filter to sort targets by name or status.

5. Select one or more targets to work with.



When you select a target you can:

- Rediscover

Direct Turbonomic to fully discover the entities that this target manages. This will rebuild the topology that is associated with this target.

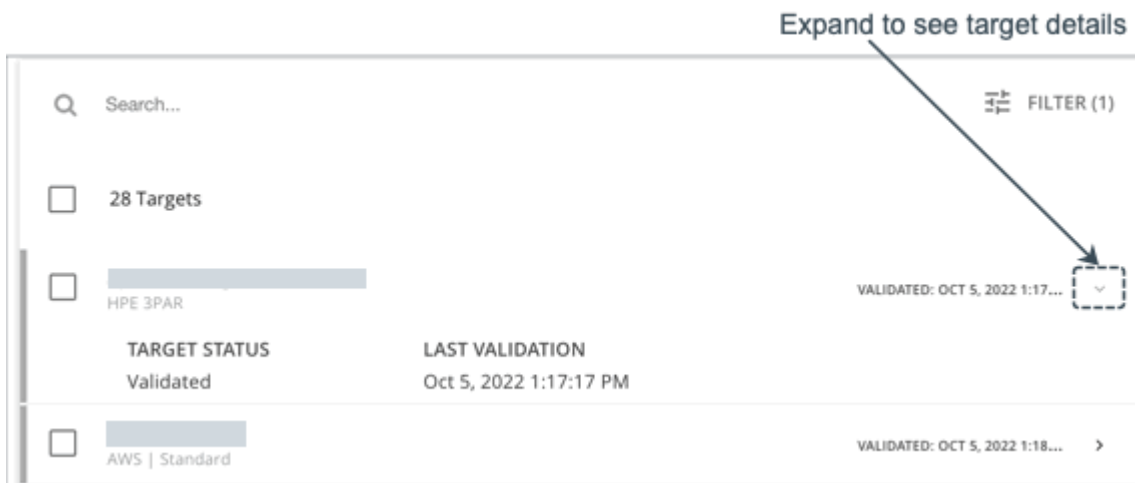
- Validate

Direct Turbonomic to validate its connection with the target. For example, if you create a new user account on the target, you can edit the target connection to use that account, and then revalidate.

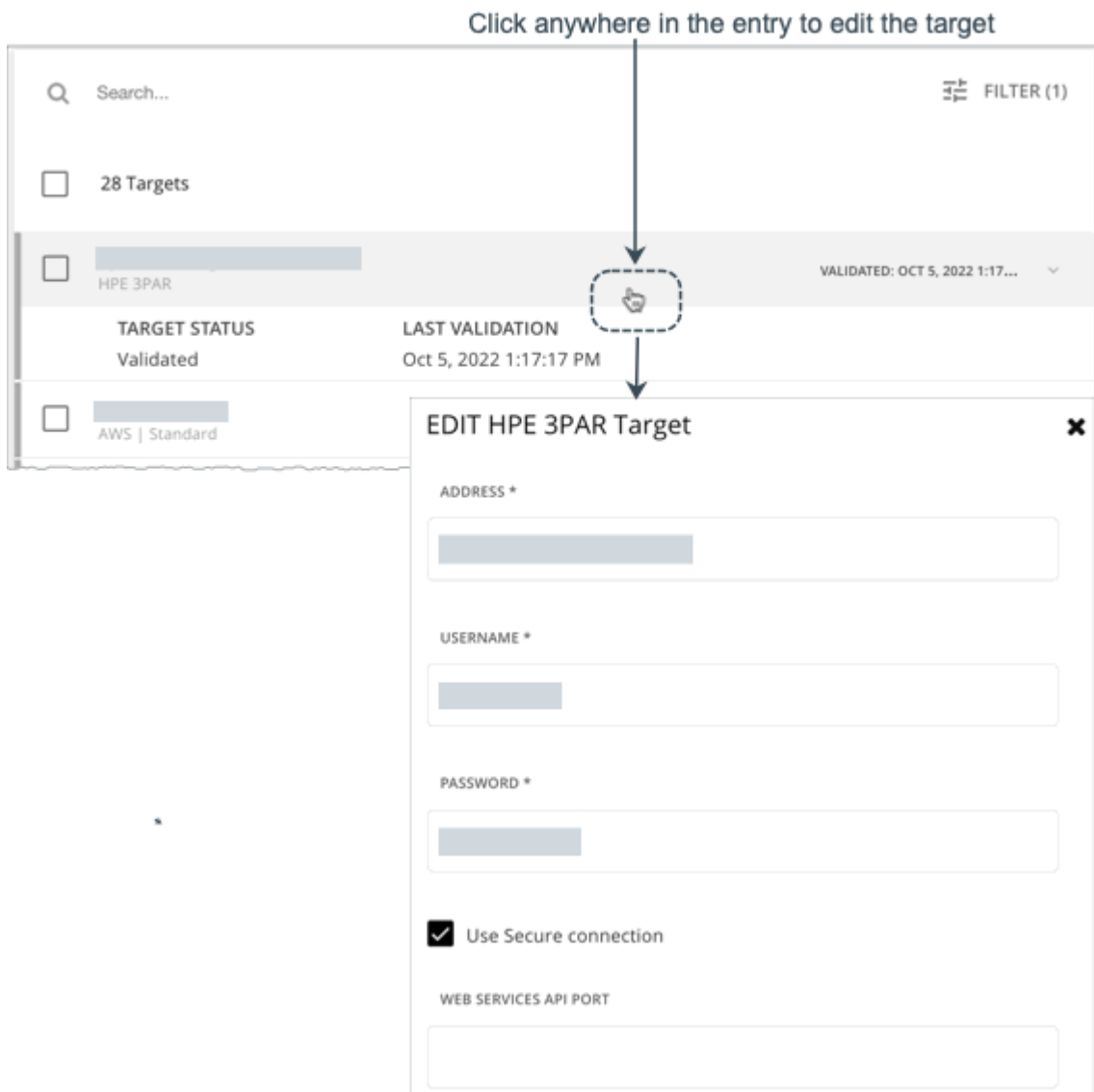
- Delete (delete icon)

When you delete a target, Turbonomic removes all the associated entities from the supply chain.

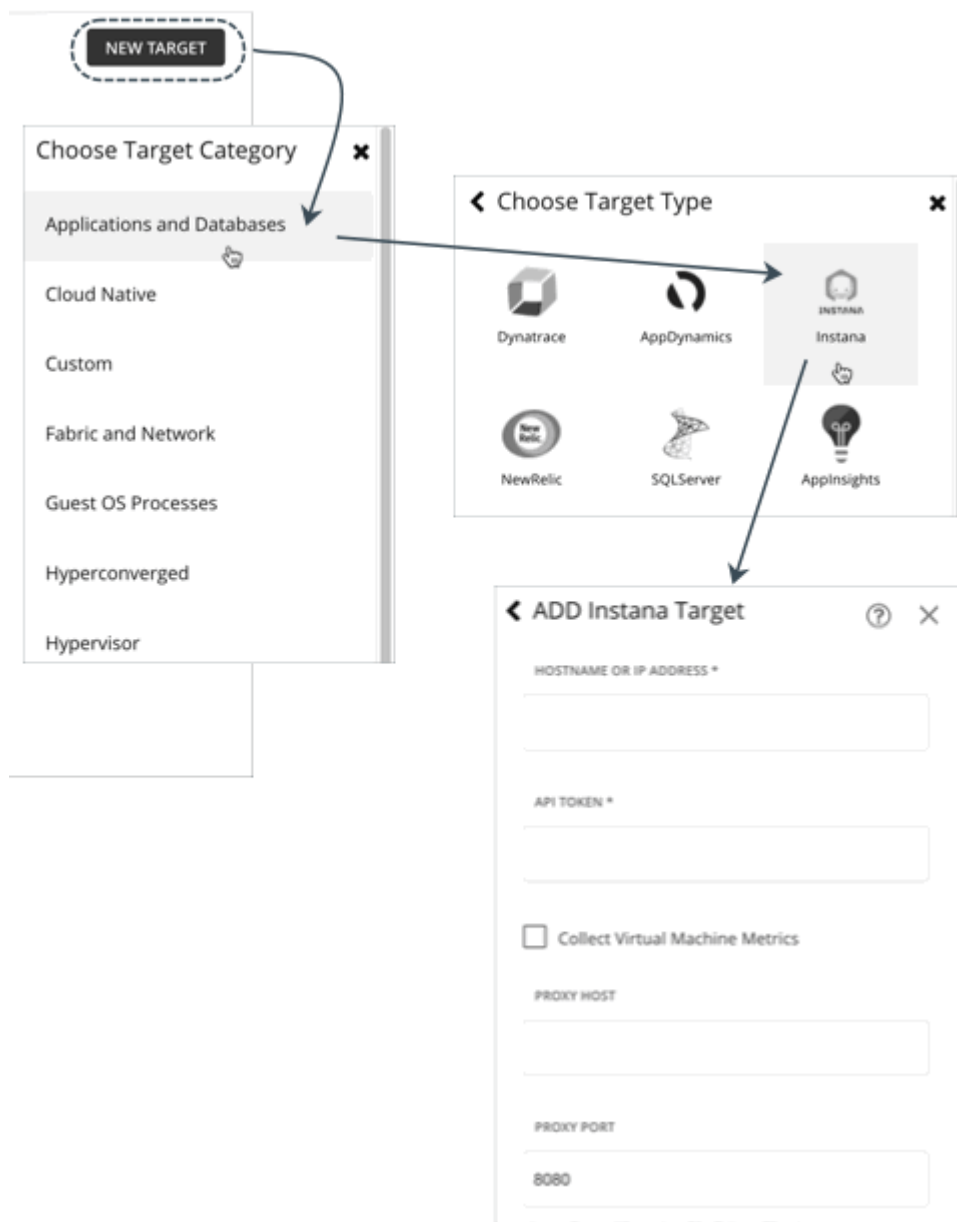
6. Expand an entry to see details.



You can also click anywhere in the entry to edit the target's configuration. For example, if you entered the wrong username or password, you can change those credentials and validate the target again.



7. Create a new target and add it to Turbonomic.



Click **New Target**, select the target category and type, and then provide the address and credentials for that target.

After you add the target, the Target Configuration page refreshes to show the current validation status.

- Validating  
Validation is in progress.
- Validated  
Validation was successful. Turbonomic can now monitor the target and will start discovering the entities that the target manages.
- Validation Failed  
Validation was unsuccessful. Expand the target to see additional information.

## Configuring Targets - Enhanced View

In the enhanced target settings experience, you can easily view targets in a tabular form with improved search, sort, and filtering capabilities. Targets with health issues are easily identified alongside

information on who changed the target configuration and when it was changed. The Target Details page for some target integrations communicate the stages of validation and discovery, as well as any encountered error conditions to aid in troubleshooting your target configurations. Related targets are also shown where applicable. Follow the instructions below to configure your target using the new view.

**Note:**

To switch to the old target experience, navigate to **Settings > Target Configuration** and turn off **New View** at the top-right section of the page. For information on the standard view, see [Configuring Targets](#).

To configure a target, you will choose the target type, specify the target's address or key, and then provide credentials to access the target. Turbonomic then discovers and validates the target, and then updates the supply chain with the entities that the target manages.

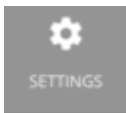
**Note:**

Turbonomic regularly checks the status of your targets. If target discovery or validation fails, the Target Configuration page updates the status. Under some circumstances, the target can become discoverable or valid again, but the status does not update. In this case, select the target and then click **Rediscover** or **Validate**.

For a list of supported targets and configuration requirements, see [Target Configuration](#).

## Configuring a target using the enhanced view

1. Navigate to the Settings page.



Click to navigate to the Settings page. From there, you can perform a variety of Turbonomic configuration tasks.

2. Choose Target Configuration.



Click to navigate to the Target Configuration Page.

3. Review the list of targets.

A screenshot of the "Target Configurations" page. At the top, there is a header with a target icon, the text "Target Configurations", a "NEW VIEW" toggle switch, and a "NEW TARGET" button. Below the header is a table with columns: Target Name, Target Type, Health, Status, Last Modified, and Details. The table contains five rows of target data. On the left side, there is a sidebar with a list of target categories: All Targets (35), Cloud Native (2), Custom (1), and Public Cloud (32). At the top right of the table area, there are "REDISCOVER" and "ADD FILTER" buttons.

This page lists all the targets that you currently have configured for Turbonomic in a sortable table. The table is sorted by target health by default. You can inspect or edit these targets, or add a new target.

4. Filter the list of targets.

Target Configurations NEW VIEW NEW TARGET

---

All Targets (35) **A**

Cloud Native (2)

Custom (1)

Public Cloud (32)

All Targets REDISCOVER

🔍 Type to search **B** **C** ADD FILTER

| <input type="checkbox"/> | Target Name | Target Type | Health ↑ | Status                            | Last Modified | Details |
|--------------------------|-------------|-------------|----------|-----------------------------------|---------------|---------|
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Failed to connect to probe. Check | 2 months ago  | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: HttpC | a month ago   | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: Azure | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | 2 months ago  | ✍       |

For a long list of targets, you can:

- a. Filter targets by target type.
- b. Use Search to filter targets by Target Name using a text string (partial matching is supported).
- c. Use Filter to filter targets by status (for example, only show validated targets). You can also use Filter to filter targets by target type or health.

5. Select one or more targets to work with.

Target Configurations NEW VIEW NEW TARGET

---

All Targets (35)

Cloud Native (2)

Custom (1)

Public Cloud (32)

All Targets REDISCOVER

🔍 Type to search ADD FILTER

| <input type="checkbox"/> | Target Name | Target Type | Health ↑ | Status                            | Last Modified | Details |
|--------------------------|-------------|-------------|----------|-----------------------------------|---------------|---------|
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Failed to connect to probe. Check | 2 months ago  | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: HttpC | a month ago   | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: Azure | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | 2 months ago  | ✍       |

When you select a target you can:

- Rediscover

Direct Turbonomic to fully discover the entities that this target manages. This will rebuild the topology that is associated with this target.

- Delete

When you delete a target, Turbonomic removes all the associated entities from the supply chain.

6. View the target details by clicking the icon under the **Details** column.

Target Configurations NEW VIEW NEW TARGET

---

All Targets (35)

Cloud Native (2)

Custom (1)

Public Cloud (32)

All Targets REDISCOVER

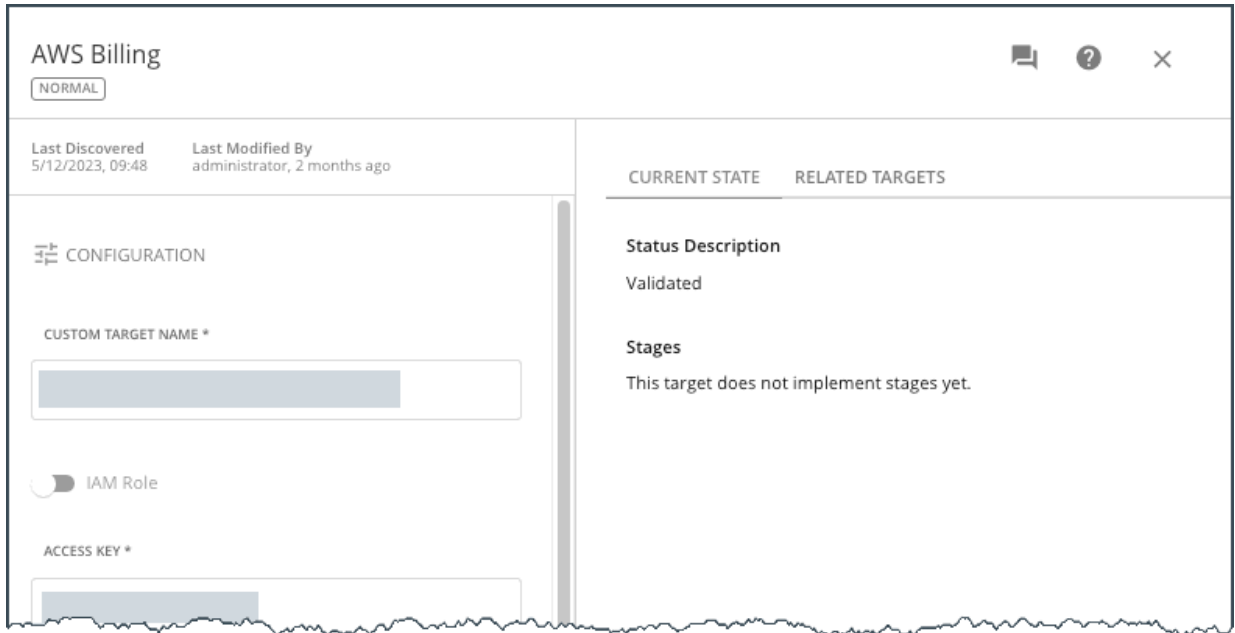
🔍 Type to search ADD FILTER

| <input type="checkbox"/> | Target Name | Target Type | Health ↑ | Status                            | Last Modified | Details |
|--------------------------|-------------|-------------|----------|-----------------------------------|---------------|---------|
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Failed to connect to probe. Check | 2 months ago  | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: HttpC | a month ago   | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | CRITICAL | Discovery failed: CRITICAL: Azure | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | N/A           | ✍       |
| <input type="checkbox"/> | [REDACTED]  | [REDACTED]  | NORMAL   | Validated                         | 2 months ago  | ✍       |

Use the target details page to view or edit target details. In addition to target details, this page shows the last discovered and last modified dates, target health, state, and any related targets.

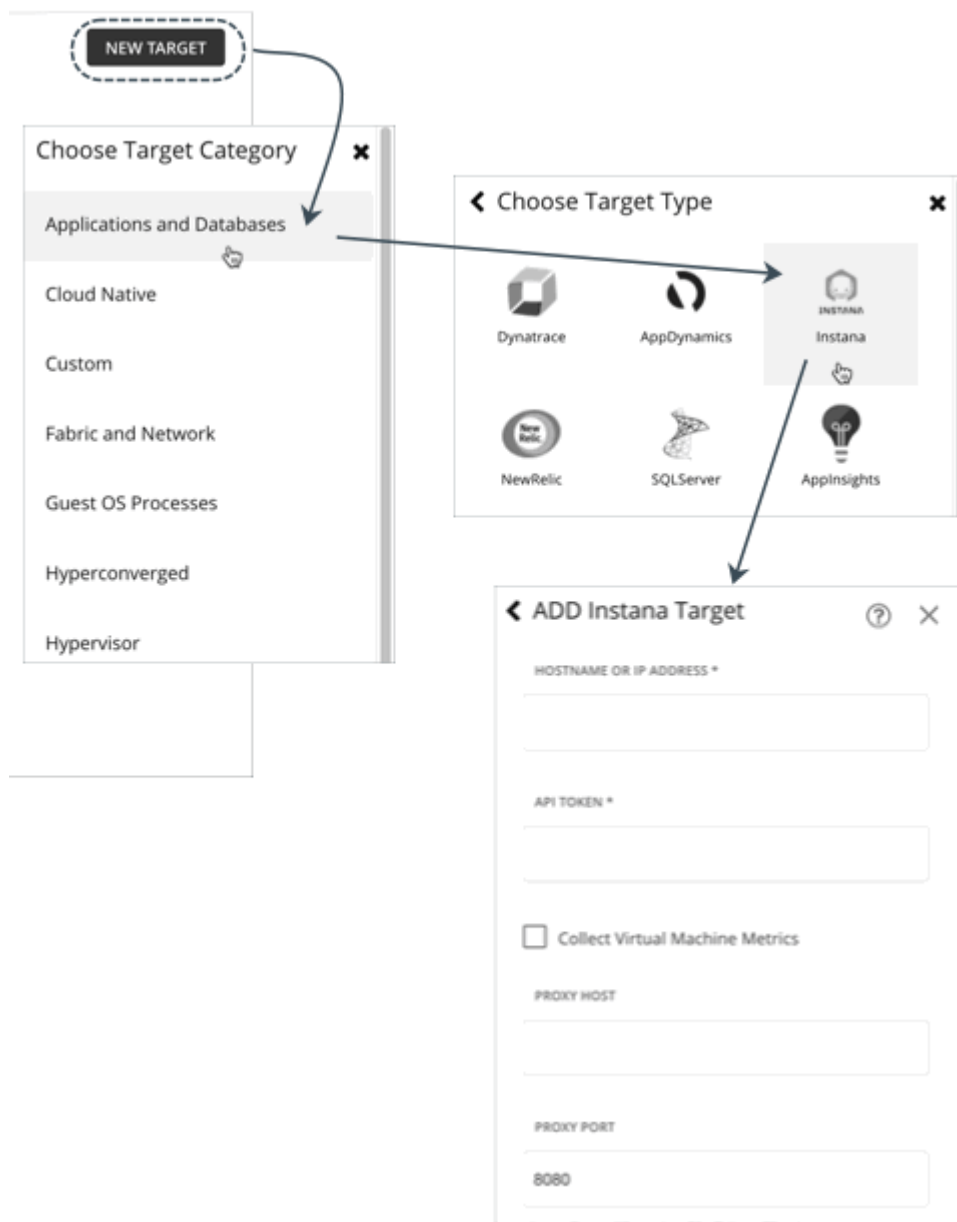
**Note:**

A value of "N/A" in the Last Modified column indicates that the target is a derived target, and therefore the details cannot be edited.



7. Create a new target and add it to Turbonomic.





Click **New Target**, select the target category and type, and then provide the address and credentials for that target.

After you add the target, the Target Configuration page refreshes to show the current validation status.

- Validating

Validation is in progress.

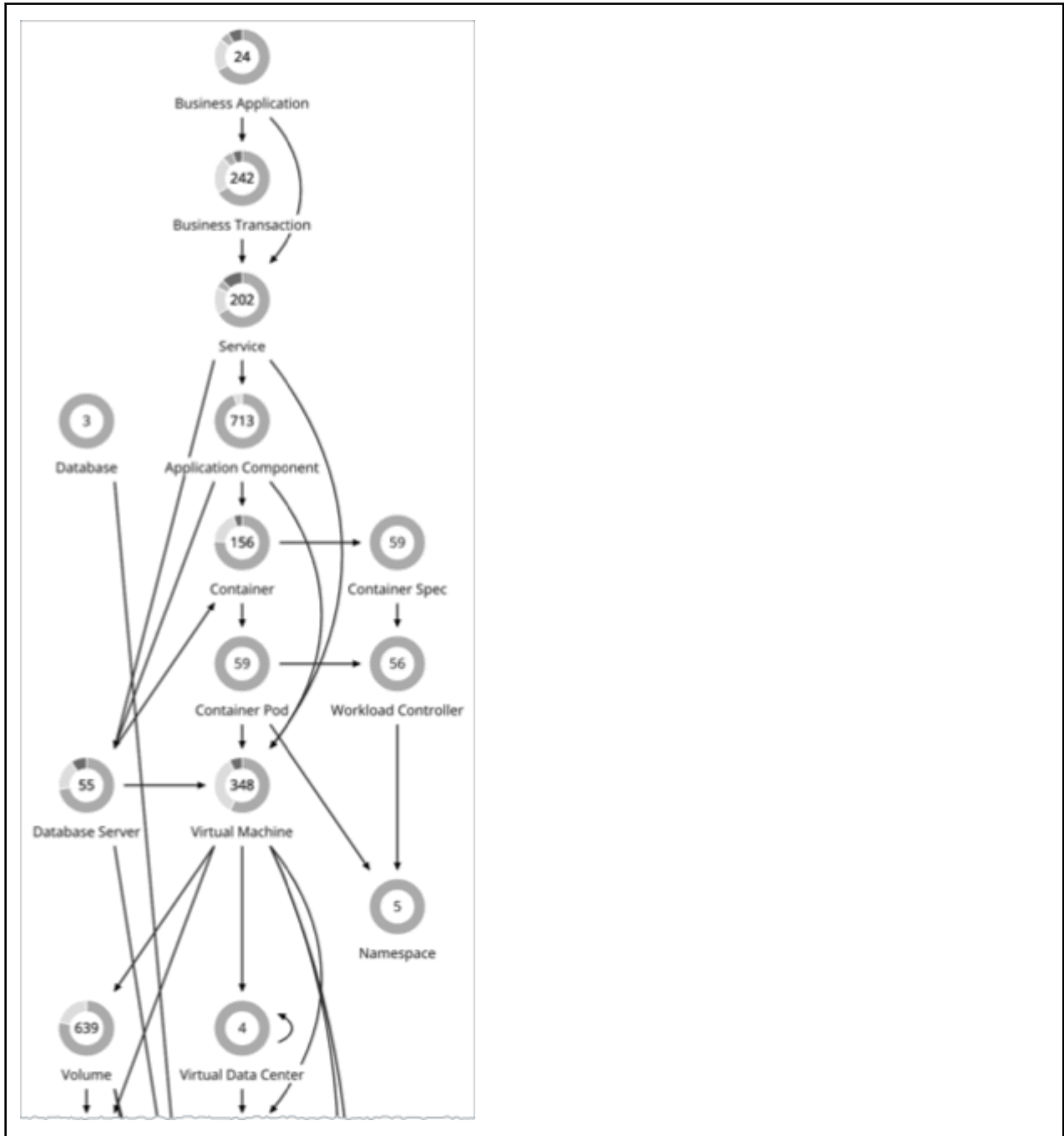
- Validated

Validation was successful. Turbonomic can now monitor the target and will start discovering the entities that the target manages.

- Validation Failed

Validation was unsuccessful. Expand the target to see additional information.

## Supply Chain of Entities



To perform Application Resource Management, Turbonomic models your environment as a market of buyers and sellers linked together in a supply chain. This supply chain represents the flow of resources from the datacenter, through the physical tiers of your environment, into the virtual tier and out to the cloud. By managing relationships between these buyers and sellers, Turbonomic provides closed-loop management of resources, from the datacenter, through to the application.

### Reading the Supply Chain

By looking at the Supply Chain, you can see:

- How many entities you have on each tier

Each entry in the supply chain gives a count of entities for the given type.

- The overall health of entities in each tier

The ring for each entry indicates the percentage of pending actions for that tier in the datacenter. Ring colors indicate how critical the actions are - Green shows the percentage of entities that have no actions pending. To get actual counts of pending actions, hover on a ring to more details.

- The flow of resources between tiers

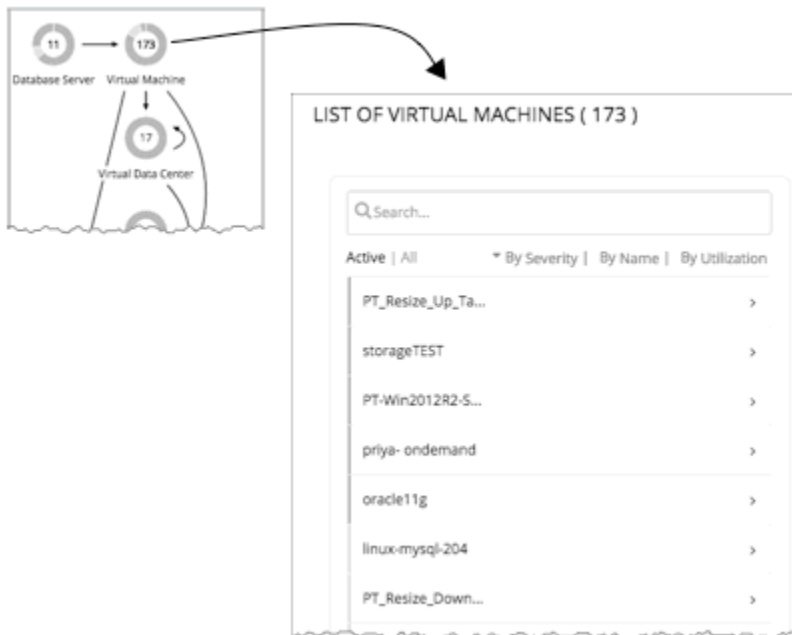
The arrow from one entry to another indicates the flow of resources. For example, the Virtual Machine entry has arrows to Hosts and to Storage. If the VMs are running in a Virtual Data Center, it will have another arrow to that as well. This means that your VMs consume resources from hosts, storage, and possibly from VDCs.

## Listing Entities From the Home Page

The Supply Chain shows the relationships of entities in your environment. When you're on the **Home Page** with a global scope, the supply chain filters its display according to the view you have chosen:

- APPLICATIONS – All your [Business Applications](#)
- ON-PREM – All your on-prem entities
- CLOUD – All your entities on the public cloud

To see a list of entities, click an entity tier in the Supply Chain.



## Working With a Scoped View

By default, the **Home Page** shows a Global view of your environment. To drill down into specifics of your environment, you can set a scope to your Turbonomic session. A scoped view shows details about the specific entities in that scope.

Once you have set a scope, you can use the Supply Chain to zoom in on a related tier to see details about the entities on that tier.

If you find the current scope to be useful, you can save it as a named group. Using named groups is an easy way to return to different scopes that you have saved.

## Scoping the Turbonomic Session

The default scope for the **Home Page** shows an overview of the global environment. What if you want to focus on less than the global environment? Assume you are responsible for a subset of workloads in your environment. This could be:

- Workloads managed on a single host cluster
- The workloads in a single datacenter
- A custom group of workloads you have created in Turbonomic

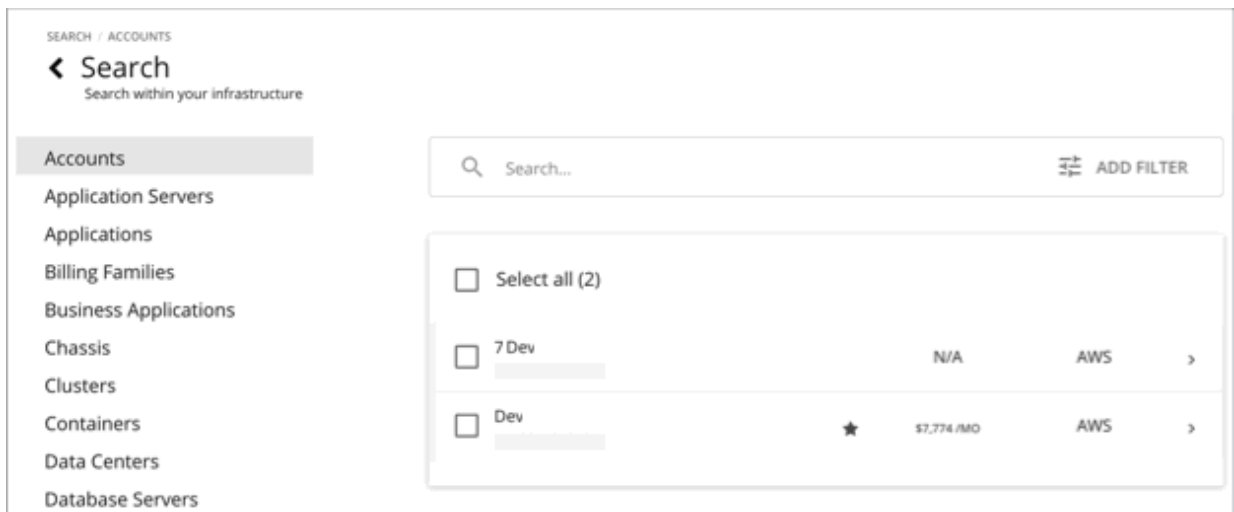
It's easy to set the session scope so that Turbonomic zooms in on the part of the environment that you want to inspect. Once you set the scope, you can get a quick picture of system health for that scope. If you find a certain scope to be useful, you can save it as a named group that you can return to later.

1. Navigate to the Search Page.



Click to navigate to the Search Page. This is where you can choose the scope you want.

2. Choose the type of entities to search.



In the Search Page, choose a type of entities that you want to search through. Find the list of entity types on the left. Select **All** to search the complete environment. Or you can focus on entities by type, by groups, or by clusters. When you select an entity type, the page updates to show all entities of that type.

3. Use **Search** to filter the listing.

For example, if you're showing **All** and you search for "Development", then you will see all clusters, groups, and entities with "Development" in their names.

Search for "Development" to filter the list



4. Expand an entry to see details.

For example, expand a group or an entity to see utilization details and pending actions.

**Note:**

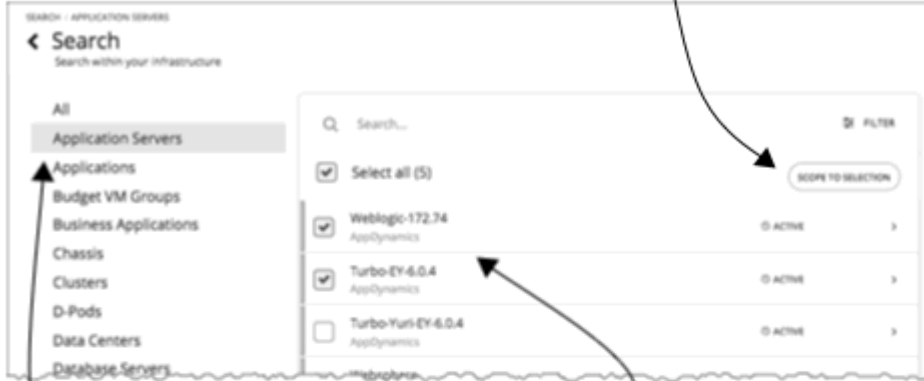
For hosts in the public cloud, utilization and capacity for host and datacenter resources don't affect Turbonomic calculations. When you expand an entry for a public cloud host, the details do not include information for these resources.

Click to show/hide details

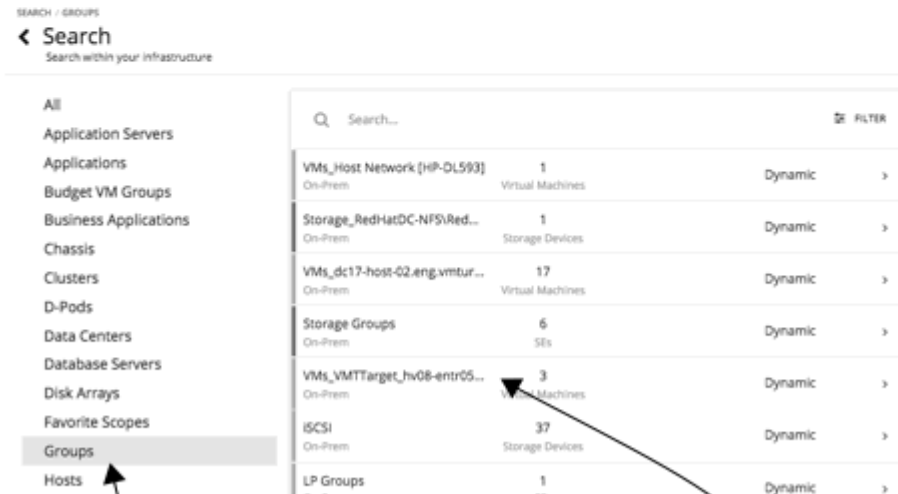


5. Select one or more entries to set the focus of the **Home Page**.

Click to set the scope you have selected



Choose an entity type, and set the scope to one or more of those entities

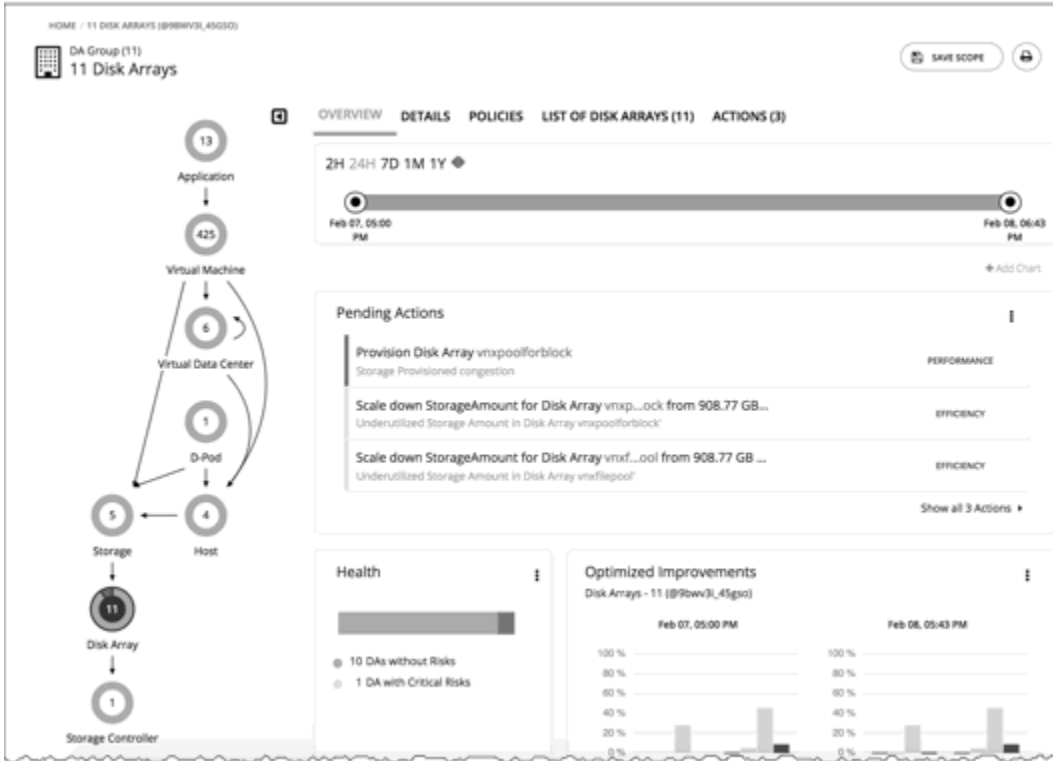


For different types of groups, click to set a single group as your scope

If you choose a category of entities to limit the list, then you can select one or more of the entities for your session scope. After you select the entities you want to include in your scope, click **SCOPE TO SELECTION** to set the session scope to those entities.

If you choose Groups or Clusters, then you can select a single entry to set the scope for your session. When you select an entry in the list, that sets the focus of the **Home Page**. For example, if you select a cluster in the **Search** listing, you set the **Home Page** focus to that cluster. Use the **Home Page** breadcrumb to set a different scope, or you can return to **Search** and set a different scope from there.

# Overview Charts



The Overview Charts show your environment's overall operating health for the current session scope. A glance at the Overview gives you insights into service performance health, overall efficiency of your workload distribution, projections into the future, and trends over time.

The charts in this view show data for the current scope that you have set for the Turbonomic session. For the global scope, the charts roll up average, minimum, and peak values for the whole environment. When you reduce the scope (for example, set the scope to a cluster), the charts show values for the entities in that scope.

Some charts included in this view are:

- Pending Actions

See all the actions that are pending for the current scope.

- Health

Quickly see the health of the entities in this scope- How many entities have risks, and how critical the risks are.

- Optimized Improvements

A comparison of utilization in your environment before executing the pending actions, and then after.

- Capacity and Usage

This chart lists resources that are used by the current scope of entities, showing utilization as a percentage of the capacity that is currently in use.

- Multiple Resources

See the utilization over time of various resources that are used by the current scope of entities.

- Top Entities

For example, Top Virtual Machines. These charts list the top consumer entities in the current scope.

- Risks Avoided

Each action addresses one or more identified risks or opportunities in your environment. This chart shows how many risks have been addressed by the executed actions.

- Accepted Actions

This chart shows how many actions have been executed or ignored, and whether they have been executed manually or automatically.

### What You Can Do:

- Set scope: See [Scoping the Turbonomic Session](#)
- Create new charts: See [Creating and Editing Chart Widgets](#)

### Setting Chart Focus

The charts update to reflect the focus that you have set for your viewing session. While viewing the Overview Charts, you can set the focus in different ways:

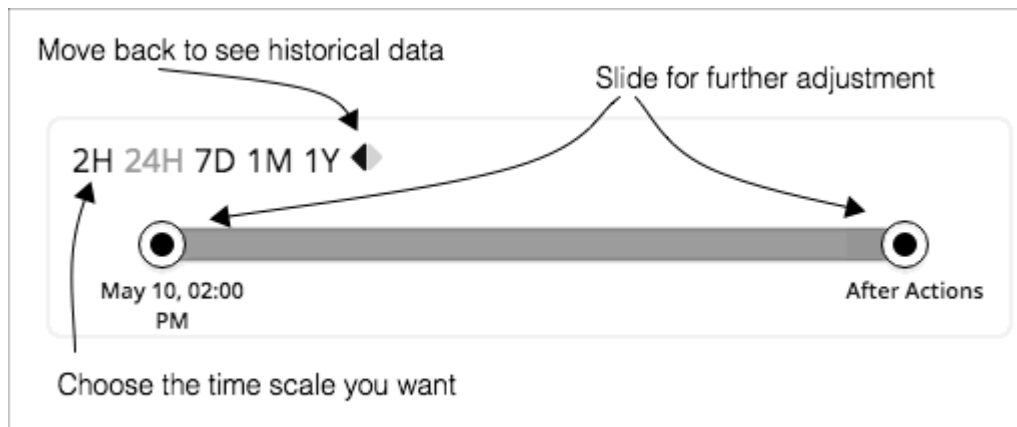
- Set Supply Chain Focus

Choose a tier in the supply chain to set the view focus - see [Navigating With the Supply Chain](#)

- Set Scope

Use **Search** to set the scope of the viewing session - see [Scoping the Turbonomic Session](#)

### Chart Time Frame



You can set a time frame from recent hours to the past year, and set that to the charts in the view. Use the Time Slider to set specific start and end times within that range. The green section in the slider shows that you can set the time range to include a projection into the future. For this part of the time range, charts show the results you would see after you execute the current set of pending actions.

For most charts, you can also configure the chart to hard-code the time range. In that case, the chart always shows the same time scale, no matter what scale and range you set for the given view.

Note that Turbonomic stores historical data in its database. As you run Turbonomic in your environment for more time, then you can set a time range to show more history.

### Details View

The Details View shows more details about the entities in your session scope. These charts focus on the utilization of resources by these entities, so you can get a sense of activity in that scope over time.





## What You Can Do:

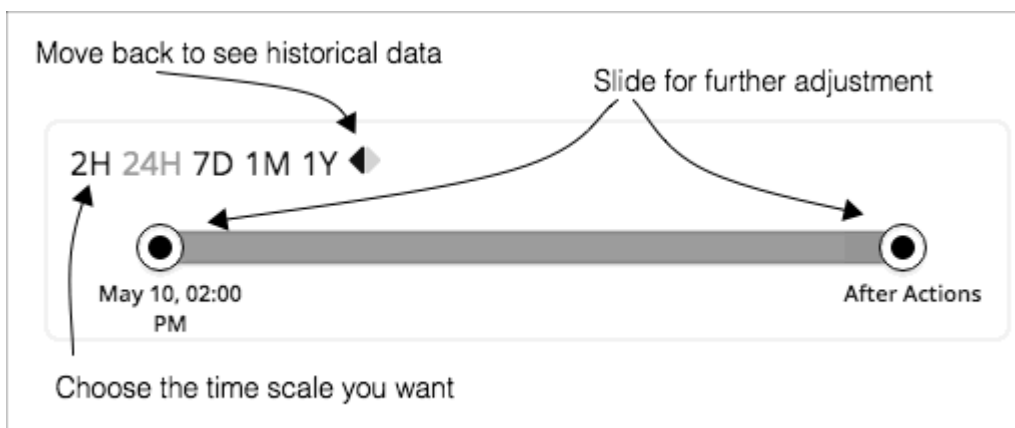
- Set scope: See [Scoping the Turbonomic Session](#)
- Create new charts: See [Creating and Editing Chart Widgets](#)

## Setting Chart Focus

The charts update to reflect the focus that you have set for your viewing session. While viewing the Overview Charts, you can set the focus in different ways:

- Set Supply Chain Focus
  - Choose a tier in the supply chain to set the view focus - see [Navigating With the Supply Chain](#)
- Set Scope
  - Use **Search** to set the scope of the viewing session - see [Scoping the Turbonomic Session](#)

## Chart Time Frame



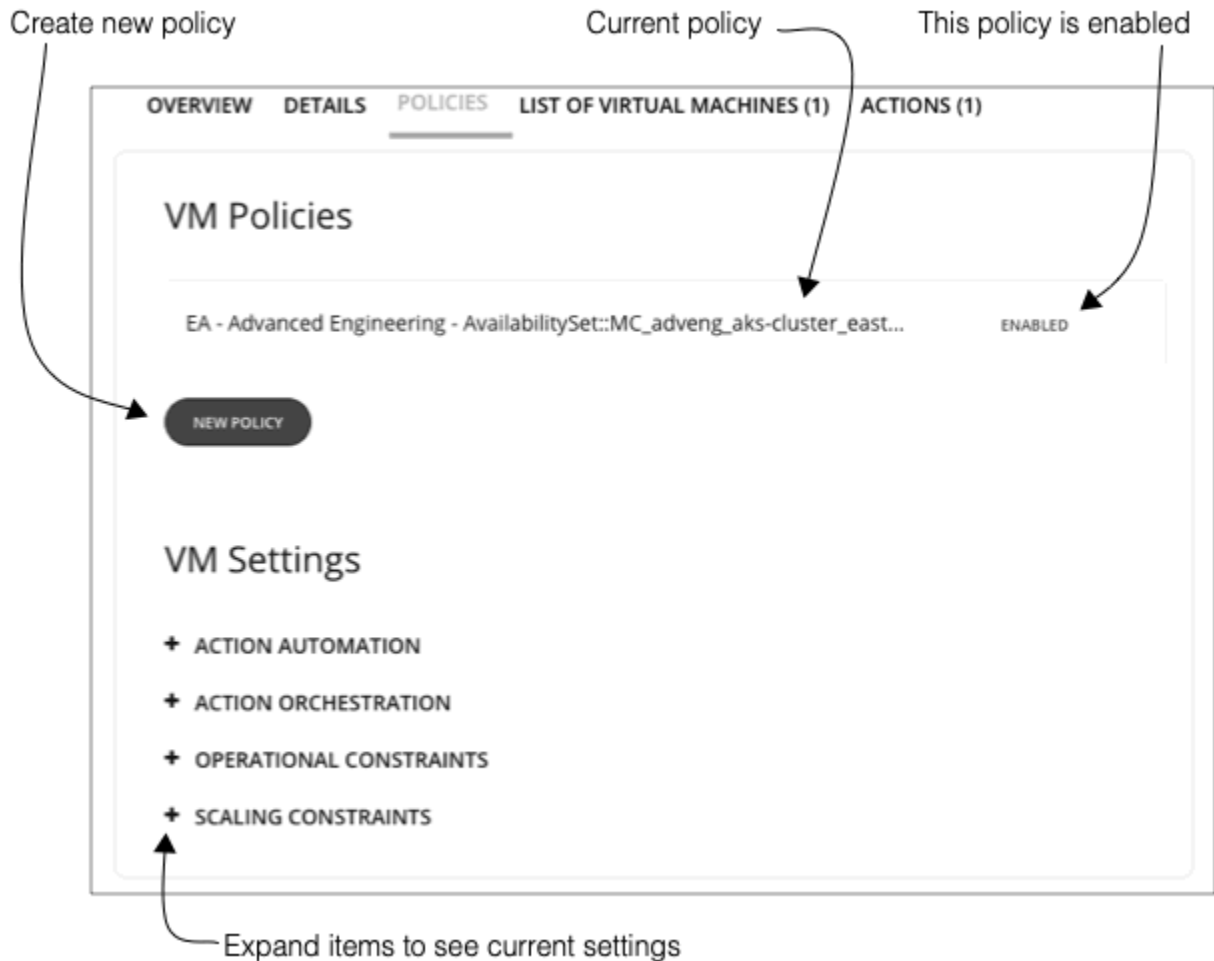
You can set a time frame from recent hours to the past year, and set that to the charts in the view. Use the Time Slider to set specific start and end times within that range. The green section in the slider shows that

you can set the time range to include a projection into the future. For this part of the time range, charts show the results you would see after you execute the current set of pending actions.

For most charts, you can also configure the chart to hard-code the time range. In that case, the chart always shows the same time scale, no matter what scale and range you set for the given view.

Note that Turbonomic stores historical data in its database. As you run Turbonomic in your environment for more time, then you can set a time range to show more history.

## Scope Policies



The Policy View gives you a look at the Automation Policies that are set for the entities in the current scope. For each policy, you can see whether it has been enabled or disabled. In addition, you can create new policies and apply them to that scope.

To edit a policy, click the policy name. You can then change the policy settings, or enable/disable the policy.

To see the current policy settings, expand a settings category. For each setting, you can see which policy determines the value- Either the default policy or a custom policy that has been applied to this scope.

When you create a new policy, it automatically includes the current scope. You can add other groups to the policy scope if you like. Note that you can enable more than one policy for the same scope. If two policies apply different values for the same setting, then the most conservative value takes effect.

For more information, see [Automation Policies](#).

## Entity Placement Constraints

| VM Placement Constraints |                              |                           |             |
|--------------------------|------------------------------|---------------------------|-------------|
| - PROVIDERS              |                              |                           |             |
|                          | CURRENT PLACEMENT            | OTHER POTENTIAL PLACEMENT |             |
| Host                     | dc17-host-01.eng.vmturbo.com | 4 Hosts                   | Constraints |

Click to see more details



When you drill down to a single entity, you can see details about the entity's relationships in the supply chain. This shows you which entities provide resources to this entity. When considering providers for this entity, you can see the name of each current provider, and how many alternative providers Turbonomic can choose from if the current one becomes overutilized.

Reviewing the constraints on an entity helps you understand the actions that Turbonomic recommends. If an action seems questionable to you, then you should look at the constraints on the affected entities. It's possible that some policy or constraint is in effect, and it keeps Turbonomic from recommending a more obvious action.

### Experimenting With Placement Constraints

For each provider or consumer in the list, you can open a **Constraints** fly-out that gives more details about limits on the current element's supply chain relationships.

For example, assume the **PROVIDERS** list shows your VM's **CURRENT PLACEMENT** is on Host A, and for **OTHER POTENTIAL PLACEMENT** you see that Turbonomic can choose from 4 hosts. When you click **Constraints**, the flyout displays a list of host constraints that currently result in the four potential hosts for this VM.

**Host Constraints For "Oracle11g-Win-172.32"** ✕

When you add constraints, you limit the placement decisions Turbonomic can make for your VM. Remove unnecessary constraints so Turbonomic can discover more placement options.

| <input type="checkbox"/> | CONSTRAINT TYPE          | SCOPE NAME                            | SOURCE     | POTENTIAL HOSTS |
|--------------------------|--------------------------|---------------------------------------|------------|-----------------|
| <input type="checkbox"/> | Cluster boundaries ⓘ     | ACM\ACM Cluster                       | vCenter    | 4 Hosts         |
| <input type="checkbox"/> | Datacenter boundaries ⓘ  | ACM                                   | vCenter    | 4 Hosts         |
| <input type="checkbox"/> | Datastore Commodity ⓘ    | QS4:ACM                               | vCenter    | 4 Hosts         |
| <input type="checkbox"/> | Network boundaries ⓘ     | NetworkCommodity/Oracle11g-Win-172.32 | Turbonomic | 12 Hosts        |
| <input type="checkbox"/> | Segmentation Commodity ⓘ | My Placement Policy                   | Turbonomic | 16 Hosts        |
| <input type="checkbox"/> | LicenseAccessCommodity ⓘ | Linux                                 | Turbonomic | 70 Hosts        |

POTENTIAL HOSTS: 4 FIND MORE PLACEMENT OPTIONS

Current count of potential providers Click to enable constraint simulations

The list information includes:

- CONSTRAINT TYPE

Most constraints are boundaries that are inherent in your environment such as a cluster boundaries or a networks, or the can be constraint rules such as discovered HA or DRS rules authored Turbonomic placement policies (sometimes called *segments*)

- SCOPE NAME

For a given rule or constraint, the scope to which it was applied.

- SOURCE

If this is a discovered constraint, the source shows the type of target that imposes this constraint. For example, for a DRS rule the source will be vCenter.

- POTENTIAL PROVIDERS

For the given constraint, how many providers that constraint allows. To see a list of the potential providers, click the POTENTIAL PROVIDERS value.

To dig deeper into how these constraints affect your entity, click **FIND MORE PLACEMENT OPTIONS**. This puts you into a *simulation mode* that you can use to experiment with changing the effective constraints. For example, you might see that a cluster boundary is limiting your placement possibilities, and you would like the option to place the current VM on other clusters. Armed with this information, you could navigate to Policies and create a Merge Cluster policy.

Use the toggles to turn off various constraints

| CONSTRAINT TYPE                                            | SCOPE NAME                            | SOURCE     | POTENTIAL HOSTS |
|------------------------------------------------------------|---------------------------------------|------------|-----------------|
| <input type="checkbox"/> Cluster boundaries                | ACM/ACM Cluster                       | vCenter    | 4 Hosts         |
| <input type="checkbox"/> Datacenter boundaries             | ACM                                   | vCenter    | 4 Hosts         |
| <input type="checkbox"/> Datastore Commodity               | Q54-ACM                               | vCenter    | 4 Hosts         |
| <input checked="" type="checkbox"/> Network boundaries     | NetworkCommodity/Oracle11g-Win-172.32 | Turbonomic | 12 Hosts        |
| <input checked="" type="checkbox"/> Segmentation Commodity | My Placement Policy                   | Turbonomic | 16 Hosts        |
| <input checked="" type="checkbox"/> LicenseAccessCommodity | Linux                                 | Turbonomic | 70 Hosts        |

Click to see the potential hosts

POTENTIAL HOSTS: 12

By turning off the 4-Host constraints, you have 12 potential hosts for this VM. Click this label to see the resulting list of providers.

Related Entities

Search...

FILTER

- dc17-host-03.eng.vmburbo.com  
vCenter | Large | ACTIVE
- hp-esx4.eng.vmburbo.com  
vCenter | Medium | FAILOVER
- hp-esx7.eng.vmburbo.com  
vCenter | Large | ACTIVE
- hp-esx8.eng.vmburbo.com  
vCenter | Large | ACTIVE
- dc17-host-01.eng.vmburbo.com  
vCenter | Large | ACTIVE

In this mode you can enable and disable different combinations of constraints. As you do, the **POTENTIAL PROVIDERS** label updates to show how many providers are available to your entity. To see the resulting list of providers, click the **POTENTIAL PROVIDERS** label.

## List of Entities

Sort the list

**44 Virtual Machines**

Search...

Expand All | Collapse All    Active | All    By Virtual CPU | By Severity | By Name | By Utilization

|             |                                       |                                          |                                    |   |
|-------------|---------------------------------------|------------------------------------------|------------------------------------|---|
| i-2-32-VM   | 52.06 GHz   5.00 %<br>CPU Provisioned | 39.99 GB   12.50 %<br>Memory Provisioned | IDLE<br>State                      | > |
|             | 2.68 TB   0.02 %<br>Storage Amount    | 2.60 GHz   0.00 %<br>Virtual CPU         | 5.00 GB   0.00 %<br>Virtual Memory |   |
| iometer VM  |                                       |                                          |                                    | < |
| i-25-39-VM  |                                       |                                          |                                    | < |
| shai-test-4 |                                       |                                          |                                    | < |

Expand/collapse details for an entry

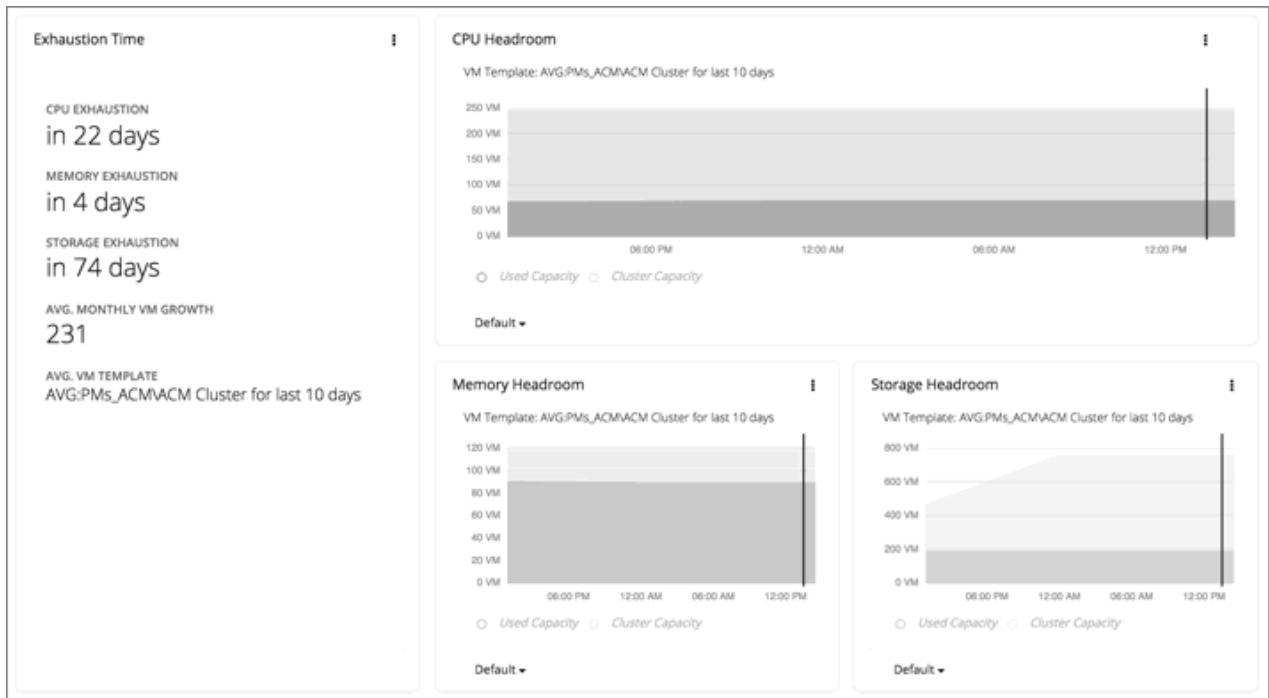
The list of entities is a quick way to drill down to details about your environment, so you can see specifics about resource consumption or state. For example, you can see the amount of capacity that has been assigned to a VM that is currently idle.

This list always updates to reflect the focus you have selected in the Supply Chain Navigator. When you select an entity type in the supply chain, the entities list updates to show the entities of that type for your current scope. For example, select Host to see a list of hosts in the current scope. For more information, see [Navigating With the Supply Chain](#)



To zoom in on a specific entity, you can click its name in the Entities List. This sets the scope to that specific entity. To return to the previous scope, use the browser's **Back** button.

## Viewing Cluster Headroom



Cluster headroom shows you how much extra capacity your clusters have to host workloads. When you set the scope to a cluster, the **Home Page** then includes charts that show headroom for that cluster, as well as time to exhaustion of the cluster resources.

To view cluster headroom:

1. Navigate to the Search page.
2. Choose the Clusters category.
3. Select the cluster you want to view.
4. When the **Home Page** displays, scroll down to show the headroom charts.

Make sure you have selected the Host tier in the supply chain navigator.

To calculate cluster capacity and headroom, Turbonomic runs nightly plans that take into account the conditions in your current environment. The plans use the Economic Scheduling Engine to identify the optimal workload distribution for your clusters. This can include moving your current VMs to other hosts within the given cluster, if such moves would result in a more desirable workload distribution. The result of the plan is a calculation of how many more VMs the cluster can support.

To calculate VM headroom, the plan simulates adding VMs to your cluster. The plan assumes a certain capacity for these VMs, based on a specific VM template. For this reason, the count of VMs given for the headroom is an approximation based on that VM template.

To specify the templates these plans use, you can configure the nightly plans for each cluster. For more information, see [Configuring Nightly Plans](#)



---

## Chapter 7. Target Configuration

A target is a service that performs management in your virtual environment. Turbonomic uses targets to monitor workload and to execute actions in your environment. Target Configuration specifies the ports Turbonomic uses to connect with these services. You must install Turbonomic on a network that has access to the specific services you want to set up as targets.

For each target, Turbonomic communicates with the service via the management protocol that it exposes — The REST API, SMI-S, XML, or some other management transport. Turbonomic uses this communication to discover the managed entities, monitor resource utilization, and execute actions.

**Note:**

Turbonomic does not support duplicate instances of the same target. When configuring targets, you must not configure two or more targets to the same address in your environment. For example, you must not configure two different targets to the same AWS account, nor two targets to the same vCenter Server instance.

If you do configure duplicate targets, then actions can fail to execute with an error that begins: `AnalysisException occurred...`

To resolve this issue, identify the duplicate targets, and delete them until you have only one target for each address.

**End-of-life (EOL) notice:** When a specific release or version of an integration partner technology reaches end-of-life (EOL) or its end of support date, Turbonomic no longer provides support for that version. Turbonomic follows integration partners' official EOL timeline for version support. Targeting a non-supported version, or one that is no longer supported by the vendor, is at your own risk.

You can assign instances of the following technologies as Turbonomic targets. Note that SaaS servers currently do not support WebSphere and vSAN.

- Applications and Databases
  - Apache Tomcat 7.x, 8.x, and 8.5.x
  - AppDynamics 4.1+
  - AppInsights
  - Datadog
  - Dynatrace 1.1+
  - IBM WebSphere Application Server 8.5+
  - Instana release-242 or later
  - JBoss Application Server 6.3+
  - JVM 6.0+
  - SQL Server 2012, 2014, 2016, 2017, 2019, and 2022
  - MySQL 5.7.x
  - NewRelic
  - Oracle 11g R2, 12c, 18c, and 19c
  - Oracle WebLogic 12c
- Cloud Native
  - Kubernetes, including any compliant k8s distribution (Rancher, Tanzu, open source, etc.)
  - Cloud-hosted k8s services (AKS, EKS, GKE, IBM, Cisco IKS, ROKS, ROSA, etc.)
  - Red Hat OpenShift 3.11 and higher (OCP 4.x)
- Fabric and Network

- Cisco UCS Manager 3.1+
- HPE OneView 3.00.04
- Guest OS Processes
  - SNMP
  - WMI: Windows versions 8 / 8.1, 10, 2008 R2, 2012 / 2012 R2, 2016, 2019 and 7
- Hyperconverged
  - Cisco HyperFlex 3.5
  - Nutanix Community Edition
  - VMware vSAN
- Hypervisors
  - IBM PowerVM 8, 9, 10
  - Microsoft Hyper-V 2008 R2, Hyper-V 2012/2012 R2, Hyper-V 2016, Hyper-V 2019
  - VMware vCenter 7.0 and 8.0
- Orchestrator
  - ActionScript
  - Flexera One
  - ServiceNow
- Private Cloud
  - Microsoft System Center 2012/2012 R2 Virtual Machine Manager, System Center 2016 Virtual Machine Manager, and System Center Virtual Machine Manager 2019
- Public Cloud
  - Amazon AWS
  - Amazon AWS Billing
  - Google Cloud Platform (GCP)
  - GCP Billing
  - Microsoft Azure Service Principal
  - Azure Billing
  - Microsoft Enterprise Agreement
- Storage
  - EMC ScaleIO 2.x and 3.x
  - EMC VMAX using SMI-S 8.1+
  - EMC VPLEX Local Architecture with 1:1 mapping of virtual volumes and LUNs
  - EMC XtremIO XMS 4.0+
  - HPE 3PAR InForm OS 3.2.2+, 3PAR SMI-S, 3PAR WSAPI
  - IBM FlashSystem running on Spectrum Virtualize 8.3.1.2 or later (8.4.2.0 or later recommended)
  - NetApp Cluster Mode using ONTAP 8.0+ (excluding AFF and SolidFire)
  - Pure Storage F-series and M-series arrays
- Virtual Desktop Infrastructure
  - VMware Horizon

## Transport Layer Security Requirements

Turbonomic requires Transport Layer Security (TLS) version 1.2 to establish secure communications with targets. Most targets should have TLS 1.2 enabled. However, some targets might not have TLS enabled, or

they might have enabled an earlier version. In that case, you will see handshake errors when Turbonomic tries to connect with the target service. When you go to the Target Configuration view, you will see a Validation Failed status for such targets.

If target validation fails because of TLS support, you might see validation errors with the following strings:

- No appropriate protocol

To correct this error, ensure that you have enabled the latest version of TLS that your target technology supports. If this does not resolve the issue, please contact IBM Technical Support.

- Certificates do not conform to algorithm constraints

To correct this error, refer to the documentation for your target technology for instructions to generate a certification key with a length of 2048 or greater on your target server. If this does not resolve the issue, please contact IBM Technical Support.

## Adding and Removing Targets

---

The target services your Turbonomic installation will manage appear in the Target Configuration list. You can add, remove, and edit entries in this list. Note that the target service's account must be configured with privileges that support the Turbonomic activities you want to perform. For example, the following list shows how vCenter privileges correspond to activities Turbonomic can perform:

- **Read Only** — Enables Turbonomic monitoring and simulation (what-if scenarios) only
- **vCenter Administrator** — Enables Turbonomic monitoring, simulation (what-if scenarios), and automation functions
- **Enable Datastore Browse** — Enabling this property for the account gives Turbonomic the privileges it needs to enable its storage management functionality

### Adding Targets

To add a target service, click the **Target Configuration** button, provide the requested information, and click **Apply** to validate those targets and start a new discovery.

#### Note:

As you add targets, be sure not to add duplicate entries for the same target.

Typical information you provide includes:

- Target Type — Choose among the supported technologies

After you choose the technology, then choose the specific target type for that technology. For example, for Cloud Management, you can choose AWS.

- Hostname or IP address — The address of the target service you want to add
- User Name — A valid account username for the target service
- Password — A password for the target service account

### Removing Targets

To remove a target, select the entry in the list and then click **Delete**.

## Applications and Databases Targets

---

Applications and Databases targets support domains of particular application servers that are controlled by management servers. For such managed domains you will add the management server as a target, and Turbonomic will discover the managed application servers.

#### Note:

As it manages your applications environment, Turbonomic discovers connected application components to stitch them into a supply chain of entities. For connections that are made by name and not IP address, Turbonomic makes DNS calls to resolve these names to IP addresses. This can happen during repeated discovery cycles.

## Supply Chain

Applications and Databases targets add Business Application, Business Transaction, Service, Application Component, Application Server, and Database Server entities to the supply chain. You can navigate to the associated target page to see how these entities map to the target nomenclature.

## Cisco AppDynamics

Turbonomic supports workload management of the application infrastructure monitored by AppDynamics, via adding the AppDynamics instance to Turbonomic as a target.

The Turbonomic integration with AppDynamics provides a full-stack view of your environment, from application to physical hardware. With information obtained from AppDynamics, Turbonomic is able to make recommendations and take actions to both assure performance and drive efficiency with the full knowledge of the demands of each individual application.

In its default configuration, the AppDynamics target will collect up to 1100 AppDynamics nodes within the default collection period when a proxy is used, and up to 5000 nodes when no proxy is required. Larger AppDynamics environments are expected to take longer than one cycle to collect complete data.

### Note:

For Kubernetes environments, Turbonomic stitches NewRelic, AppDynamics, Instana, and Dynatrace containerized application components into the supply chain to provide a unified view of your applications. For more information, see [Cloud Native Targets](#).

## Prerequisites

- A valid AppDynamics user account.

For all types of application instances, the service account must have the `Read Only User` role. For monitoring database instances, this user must also have the `DB Monitoring User` role.

### Note:

In newer versions of AppDynamics where these roles are available, they should be used instead:

- Applications and Dashboards Viewer
- DB Monitoring User
- Server Monitoring

To use a custom role, ensure that the role has the `View Server Visibility` permission for both applications and databases.

## AppDynamics Database Servers

AppDynamics also monitors database servers. In order for your database servers to be correctly stitched to the rest of your environment, you must:

- Enable enhanced metric collection.

For Hyper-V hosts, you must install Hyper-V Integration Services on the target VM hosting the database. For more information, please refer to the following integration services TechNet article:

<https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx>

For VMware hosts, you must install VMware Tools on the target VMs.

- Ensure that the database name in AppDynamics is resolvable to an IP address by the Turbonomic instance.

You may need to make changes to your DNS or the file `/etc/resolv.conf` on the Turbonomic instance.

## Adding an AppDynamics Target

### Note:

It is possible to monitor certain applications or database servers with both AppDynamics and Turbonomic, but this must be avoided as it will cause the entities to appear duplicated in the market.

If an application is monitored by AppDynamics, do not add it as a separate Turbonomic application target.

To add an AppDynamics instance as a target, specify:

- Hostname or IP Address

The host name or IP Address of the AppDynamics controller instance.

- Port

the port used to connect to the AppDynamics controller. By default, this is set to ports 80 (HTTP) and 443 (HTTPS).

**Note:** For SaaS-based AppDynamics instances, you must use port 443.

- Username or API Client Name@Account

Username and account ID with the necessary role(s). The format must be *Username@Tenant*, and the user must have the "Read Only User" and "DB Monitoring User" permissions. This username can be found on the "License > Account" page in AppDynamics. For OAuth authentication, the username must be a user defined as an API Client.

### Note:

The username and password cannot contain any of the following special characters:

`\ / " [ ] : | < > + = ; , ? * , ' tab space @`

- Password or Client Secret

Password for the account used to connect to the AppDynamics instance. For OAuth, this will be the client secret key.

### Note:

The username and password cannot contain any of the following special characters:

`\ / " [ ] : | < > + = ; , ? * , ' tab space @`

- Collect Virtual Machine Metrics

Whether to collect VM metrics from this target. When **Collect Virtual Machine Metrics** is turned on, Turbonomic uses the VM metrics it collects from this target, instead of the VM data that can optionally be collected from related infrastructure targets.

- Secure Connection

When checked, Turbonomic will connect via HTTPS. Make sure the required certificate is configured for use on the host.

- Use API Client (OAuth)

When checked, enables Open Authorization (OAuth) token-based authentication for the target connection.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port  
The port to use with the proxy specified above. By default, this is 8080.
- Proxy Username  
The username to use with the proxy specified above.
- Proxy Password  
The password to use with the proxy specified above.
- Secure Proxy Connection  
When checked, Turbonomic will connect to the proxy via HTTPS.

For more information about creating API client users, see the [AppDynamics Documentation](#).

## Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the target and Turbonomic.

| AppDynamics                                  | Turbonomic            |
|----------------------------------------------|-----------------------|
| Business Application                         | Business Application  |
| Business Transaction                         | Business Transaction  |
| Tier                                         | Service               |
| Node                                         | Application Component |
| Database                                     | Database Server       |
| Machine (when the machine type is Container) | Container             |
| Server                                       | Virtual Machine       |

## Monitored Resources

### Note:

The exact resources this target monitors can differ based on application type. The following list of metrics per entity includes all resources you might see.

For a given VM, the resources you see depend on how the VM is discovered, and whether the VM provides resources for an application discovered by this target:

- If the VM hosts an application that is discovered through this target, then you will see VM metrics discovered through this target.
- If the VM is discovered through a different target, and it does not host any application discovered through this target, then you will see VM metrics discovered through that different target.
- If the VM is discovered through this target, but it does not host any application discovered through this target, then Turbonomic does not display metrics for the VM.

Turbonomic monitors the following resources for the AppDynamics supply chain:

- Business Transaction
  - Response Time  
Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions
 

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
- Business Applications
  - Response Time
 

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).
  - Transactions
 

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
- Service
  - Response Time
 

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

For Kubernetes, this is the desired *weighted average* response time of all Application Component replicas associated with a Service.
  - Transactions
 

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

For Kubernetes, this is the maximum number of transactions per second that each Application Component replica can handle.
- Application Component
  - Virtual CPU (Java, .NET, and Node.js only)
 

Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Memory (Java, .NET, and Node.js only)
 

Virtual Memory is the measurement of memory utilized by the entity.
  - Transactions
 

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Heap (Java, .NET, and Node.js only)
 

Heap is the portion of a VM or container’s memory allocated to individual applications.
  - Response Time
 

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).
  - Connections
 

Connection is the measurement of Database Server connections utilized by applications.
  - Remaining GC Capacity
 

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).
  - Threads
 

Threads is the measurement of thread capacity utilized by applications.
- Database Server
  - Transactions (SQL, MySQL, and Oracle only)

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Connections (MongoDB only)

Connection is the measurement of Database Server connections utilized by applications.

- Transaction Log (SQL only)

The utilization of the server's capacity for storage devoted to transaction logs

- DB Cache Hit Rate (SQL and Oracle only)

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Virtual Machine

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

**Note:**

To collect data, a machine agent must be present and database hardware monitoring must be enabled.

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

**Note:**

To collect data, a machine agent must be present and database hardware monitoring must be enabled.

## Actions

**Note:**

The specific actions that Turbonomic recommends can differ, depending on the processes that Turbonomic discovers.

For other application components, Turbonomic can recommend actions based on the resources it can discover for the application. For example, Node.js® applications report CPU usage, so Turbonomic can generate vCPU resize actions and display them in the user interface.

Turbonomic recommends actions for the AppDynamics supply chain as follows.

- Database Server

- Resize Database Memory (DBMem)

Recommendation only.

- Resize Connections

Recommendation only.

**Note:**

For different types of Database Servers, the AppDynamics target returns different metrics. This affects Turbonomic actions as follows:

- MySQL:

For MySQL database servers, analysis does not generate resize actions for DB Memory or Connections. The target does not discover DB Cache Hit Rate, DB Memory or Connections.

- SQL Server:

For SQL database servers, analysis does not generate resize actions for DB Memory or Connections. The target does not discover DB Memory or Connections.



- MongoDB:

For MongoDB database servers, analysis does not generate resize actions for DB Memory or Connections. The target does not discover DB Cache Hit Rate or DB Memory.

- Oracle:

For Oracle database servers, analysis does not generate resize actions for DB Memory or Connections. The target does not discover DB Memory or Connections.

## Application Insights

Turbonomic supports workload management of the application infrastructure monitored by Application Insights via the Turbonomic integration, which provides a full-stack view of your environment, from application to hosting server. With information obtained from Application Insights, Turbonomic is able to make recommendations and take actions to both assure performance and drive efficiency with the full knowledge of the demands of each individual application.

### Prerequisites

- A valid Application Insights user account with the same permissions detailed for [Microsoft Azure targets](#).

### Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes the mapping of entities between the target and Turbonomic:

| Application Insights | Turbonomic            |
|----------------------|-----------------------|
| Application          | Application Component |

### Adding an Application Insights Target

#### Note:

If an application is monitored by Application Insights or Azure, do not add it as a separate Turbonomic application target.

To add Application Insights as a target, specify:

- Unique Target Description

A user-created name that will appear in the Turbonomic UI.

- Tenant Name

The tenant associated to the Azure subscription associated to Application Insights.

- Azure Subscription ID

The ID of the Azure subscription with access to the Azure target associated to Application Insights.

- Client ID

The Client ID of the App registration that gives Turbonomic access to resources in your Azure subscription.

- Client Secret Key

The secret key for the App registration.

- Offer ID

If applicable, the Azure Offer ID related to the Azure subscription.

- Enrollment Number

If an Azure EA account, the enrollment number associated to the Azure subscription.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username to use with the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

## Stitching Application Insights Applications to the Turbonomic Environment

Each application monitored by Application Insights can have several associated applications, and each of these applications can be deployed on a different hosting server. Likewise, a hosting server may host multiple groups or partial groups of multiple applications.

In order to accurately stitch metrics from Application Insights, the host name or IP address of the hosting server must be discoverable through the Application Insights instance. For most monitored application instances, this is automatic. If the hosting server is not discoverable, Azure tags on the application can be provided to indicate the IP address or hostname.

You can provide an Azure tag in the following name : value format:

```
Turbonomic-Host-Name :
RoleInstance=hostname;RoleInstance=hostname;RoleInstance=hostname;
```

In the preceding example, RoleInstance is the name of the application instance, and hostname is the hosting server. For example: Turbonomic-Host-Name : cluster-app-a=120.120.120.10;cluster-app-b=120.120.120.11;cluster-app-c=120.120.120.12;.

**Note:** If these tags are provided, they will replace any values discovered through the Application Insights API.

## Actions

**Note:**

The specific actions that Turbonomic recommends can differ, depending on the processes that Turbonomic discovers.

For other application components, Turbonomic can recommend actions based on the resources it can discover for the application. For example, Node.js® applications report CPU usage, so Turbonomic can generate vCPU resize actions and display them in the user interface.

\

- Application Component

- Suspend VM

- Recommendation only.

- Provision VM

- Recommendation only.

## Monitored Resources

### Note:

The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the Application Insights supply chain:

- Application Component
  - Virtual CPU  
Virtual CPU is the measurement of CPU utilized by the entity.
  - Transactions  
Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Response Time  
Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

## Datadog

The Turbonomic-Datadog integration combines application performance data with infrastructure data that Turbonomic collects from Datadog to generate full-stack aware, automatable actions that optimize application performance and reduce cost. By using IBM Turbonomic analytics and automation capabilities, you can dynamically allocate resources to ensure optimal application performance, while you minimize infrastructure costs and comply with licensing and other operational policies.

### Prerequisites

- A valid Datadog user account with read-only API and Application Key access

### Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the target and Turbonomic:

| Datadog                                                                   | Turbonomic            |
|---------------------------------------------------------------------------|-----------------------|
| Tags, which are configured in Datadog to define your business application | Business Application  |
| Resources                                                                 | Business Transaction  |
| Service                                                                   | Service               |
| Apps                                                                      | Application Component |
| Container                                                                 | Container             |
| Host                                                                      | Virtual Machine       |

### Supported Applications

Turbonomic discovers the following application types and associated commodities from the Datadog target:

| Application Type             | Commodities                                                                           |
|------------------------------|---------------------------------------------------------------------------------------|
| Stand-Alone Java Application | Virtual CPU, Virtual Memory, Heap, Remaining GC Capacity                              |
| Tomcat                       | Virtual CPU, Virtual Memory, Response Time, Transactions, Heap, Remaining GC Capacity |

## Adding a Datadog Target

### Note:

If an application is monitored by Datadog, do not add it as a separate Turbonomic application target.

To add a Datadog instance as a target, specify the following values:

- Datadog URL  
The address of the Datadog server.
- API Key  
The API Key that is provided by the Datadog platform. For more information, see [Datadog API Keys](#).
- Application Key  
The Application Key that is provided by the Datadog platform. For more information, see [Datadog API Keys](#).
- Collect Virtual Machine Metrics  
Whether to collect VM metrics from this target. When **Collect Virtual Machine Metrics** is turned on, Turbonomic uses the VM metrics it collects from this target, instead of the VM data that can optionally be collected from related infrastructure targets.
- Business Application Datadog Tag  
The Datadog tag that is defined in the configuration file to identify Business Applications. The tag name must begin and end with alphabetical characters and can contain hyphens, dots, and underscores. Custom tag values that have a matching tag name with the target configuration value are identified as a valid Business Application.
- Proxy Host (Optional)  
The IP address of the Proxy Host.
- Proxy Port  
The port that is required by the proxy.
- Proxy Username  
The username that is required by the specified proxy.
- Proxy Password  
The password that is required by the specified proxy.
- Secure Proxy Connection  
When this option is checked, Turbonomic connects to the proxy over HTTPS.

## Monitored Resources

### Note:

The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the Datadog supply chain:

- Business Application

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Note:**

- For the Datadog target to discover and link Business Applications with the underlying entities in the supply chain, you must configure application tags at the host level in the main Datadog yaml file. The Application Tag name should match the Business Application Datadog Tag name defined in the Target Configuration page.

- Business Transaction

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second usage of the transactions that are allocated to an entity.

- Service

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second usage of the transactions that are allocated to a given entity.

- Note:**

- For the Datadog target to link services with the underlying entities in the supply chain, you must configure the `service` and `env` tags at the host level in the main Datadog yaml file.

- Application Component

- Virtual CPU (JVM Application)

- Virtual CPU is the measurement of CPU that the entity uses.

- Virtual Memory (Tomcat Application)

- Virtual Memory is the measurement of memory that the entity uses.

- Transactions

- Transaction is a value that represents the per-second usage of the transactions that are allocated to a given entity.

- Heap

- Heap is the portion of a VM or container's memory that is allocated to individual applications.

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Remaining GC Capacity

Remaining GC Capacity is the measurement of Application Component uptime that is not spent on garbage collection (GC).

- Container
  - Virtual Memory

VMem is the virtual memory utilized by a container against the memory limit. If no limit is set, node capacity is used.
  - Virtual CPU

VCPU is the virtual CPU (in mCores) utilized by a container against the CPU limit. If no limit is set, node capacity is used.
- Virtual Machine
  - Virtual CPU

Virtual CPU is the measurement of CPU that the entity uses.
  - Virtual Memory

Virtual Memory is the measurement of memory that the entity uses.

## Actions

### Note:

The actions that Turbonomic recommends might differ, depending on the application type that Turbonomic discovers and supports for the Datadog target.

Turbonomic recommends actions for the following entities and application types:

- Application Component
  - Resize Heap

Recommendation only.

### Note:

Actions to resize heap are driven by the Application Type. For Tomcat Applications, analysis only generates resize actions for the Remaining GC Capacity commodity. For JVM applications, analysis only generates resize actions for the Heap commodity.

## Dynatrace

Turbonomic supports discovery of applications that are managed by the Dynatrace platform. Turbonomic includes the discovered information about these applications in its calculations for VM actions.

### Note:

For Kubernetes environments, Turbonomic stitches NewRelic, AppDynamics, Instana, and Dynatrace containerized application components into the supply chain to provide a unified view of your applications. For more information, see [Cloud Native Targets](#).

## Prerequisites

- A Dynatrace Server instance

This instance must be configured to monitor applications running in your environment.

Turbonomic supports both SaaS and on-prem Dynatrace server installations.
- Managed VMs that host applications managed by Dynatrace

For Turbonomic to discover applications through Dynatrace, the applications must be running on VMs in your environment. Also, these VMs must be managed by Turbonomic targets such as hypervisors or public cloud targets.

- An API access token with the proper scopes

Turbonomic uses the API token to authenticate its calls to the Dynatrace API. This token must have permission to execute GET methods via the Dynatrace API, both Version 1 and Version 2. Generate a new generic access token with these scopes:

| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                          |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"> <li>– API V1 scopes:               <ul style="list-style-type: none"> <li>- Access problem and event feed, metrics, and topology</li> </ul> </li> <li>– API V2 scopes:               <ul style="list-style-type: none"> <li>- Read entities</li> <li>- Read metrics</li> </ul> </li> </ul> |

**Note:**

If you are updating to Turbonomic version 8.6.2 or later, from a version that is earlier than 8.6.2, you must generate a new API token for each existing Dynatrace target. Then you must enter that token in the target configuration, and validate the target.

If the target still fails to validate after you update the access token, take note of your configuration settings, delete the target, and configure the target again. Be sure to use the new API token that you have generated.

### Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The entity names that Turbonomic displays in the Supply Chain differ from the entity names that Dynatrace displays in its user interface, as follows:

| Dynatrace Naming | Turbonomic Entity                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application      | Business Application<br><b>Note:</b><br>For Dynatrace Applications, Turbonomic displays Business Application entities in the supply chain when they have been active for at least three days. |
| Service          | Service                                                                                                                                                                                       |
| Process          | Application Component, Database Server                                                                                                                                                        |
| NA               | Container                                                                                                                                                                                     |
| Host             | Virtual Machine                                                                                                                                                                               |

### Adding a Dynatrace Target

**Note:**

You can manage certain applications or database servers with both Dynatrace and Turbonomic. You should avoid such a configuration because it can cause Turbonomic to generate duplicate entities in the market.

If you manage an application via a Dynatrace server, and you configure that Dynatrace server as a Turbonomic target, then be sure you have *not* added that application as a separate application target in Turbonomic.

To add a Dynatrace server instance as a target, specify:

- Hostname or IP Address

For an on-prem installation of Dynatrace, give the host name or IP and endpoint, separated by a slash. For example, 10.10.10.10/e/b70e3eb2-e82b-4c13-a5a4-560d9865841r.

For a SaaS installation, provide the URL *without* the {IP\_ADDRESS}/e/ statement in the path. For example, ayz12745.live.dynatrace.com

- API Token

The token that Turbonomic can use to authenticate its calls to the Dynatrace API. This token must have permission to execute GET methods via the Dynatrace API V1 and V2. Refer to the Prerequisites section for more information.

- Collect Virtual Machine Metrics

Whether to collect VM metrics from this target. When **Collect Virtual Machine Metrics** is turned on, Turbonomic uses the VM metrics it collects from this target, instead of the VM data that can optionally be collected from related infrastructure targets.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the Dynatrace server via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username for the account to log into the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

Whether to use a secure connection with the proxy. When this is turned on, Turbonomic connects to the proxy through HTTPS.

## Actions

Turbonomic recommends actions for the application supply chain as follows.

- Application Component

- Resize

- Heap

- Recommended, only.

| Entity:               | Actions: |
|-----------------------|----------|
| Application Component |          |

## Monitored Resources

**Note:**



The subset of resources that Turbonomic discovers for an application depends on the application type. The following list of metrics per entity includes the full set of resources Turbonomic can discover for Dynatrace applications.

For Database Server applications, Turbonomic only discovers metrics for MySQL and SQL Server databases.

For a given VM, the resources you see depend on how the VM is discovered, and whether the VM provides resources for an application discovered by this target:

- If the VM hosts an application that is discovered through this target, then you will see VM metrics discovered through this target.
- If the VM is discovered through a different target, and it does not host any application discovered through this target, then you will see VM metrics discovered through that different target.
- If the VM is discovered through this target, but it does not host any application discovered through this target, then Turbonomic does not display metrics for the VM.

Turbonomic monitors the following resources for the Dynatrace supply chain:

- Business Application

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Service

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Application Component

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.

- Remaining GC Capacity (Java only)

- Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

- Heap (Java only)

- Heap is the portion of a VM or container's memory allocated to individual applications.

- Database Server

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.

- Database Memory (SQL and MySQL only)

Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM.

- DB Cache Hit Rate (SQL only)

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Transactions (SQL only)

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Container

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual Machine

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

## New Relic

Turbonomic supports workload management of the application infrastructure monitored by New Relic, from application instance to host. With information obtained from New Relic, Turbonomic can make recommendations and take actions to both assure performance and drive efficiency to address the demands of each individual application. For Kubernetes environments, Turbonomic stitches containerized application components into the supply chain to provide a unified view of your applications.

### Note:

For Kubernetes environments, Turbonomic stitches NewRelic, AppDynamics, Instana, and Dynatrace containerized application components into the supply chain to provide a unified view of your applications. For more information, see [Cloud Native Targets](#).

## Prerequisites

- A valid New Relic user account that includes both APM and infrastructure monitoring.

## Entity Mapping

After validating the new target, Turbonomic discovers the connected entities. The following table describes the mapping of entities between the target and Turbonomic:

| New Relic Term                             | Turbonomic Term       |
|--------------------------------------------|-----------------------|
| APM: Key Transactions                      | Business Transaction  |
| APM: Application / Service (New Relic One) | Service               |
| APM: Application Instance                  | Application Component |

| New Relic Term  | Turbonomic Term |
|-----------------|-----------------|
| Infra: Database | Database Server |
| Infra: Host     | Virtual Machine |

For VM entities

## Supported Applications

Turbonomic discovers the following application types (and associated commodities) via the New Relic target:

| Application Type | Commodities                                                                              |
|------------------|------------------------------------------------------------------------------------------|
| .NET             | Virtual CPU, Virtual Memory, Response Time, Transactions                                 |
| GO               | Virtual CPU, Virtual Memory, Response Time, Transactions                                 |
| Java             | Virtual CPU, Virtual Memory, Response Time, Transactions, Heap, Collection Time, Threads |
| Node.js          | Virtual CPU, Virtual Memory, Response Time, Transactions, Heap, Collection Time          |
| PHP              | Virtual CPU, Virtual Memory, Response Time, Transactions                                 |
| Python           | Virtual CPU, Virtual Memory, Response Time, Transactions                                 |

## Supported Databases

Turbonomic supports the following Database types and commodities:

**Note:** Database commodities are exposed only if the New Relic account used to connect to Turbonomic has a New Relic Infrastructure Pro subscription.

| Database | Commodities                                                                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MS SQL   | Cache Hit Rate, Virtual Memory, Transactions                                                                                                                                                                                       |
| MySQL    | Cache Hit Rate<br><b>Note:</b><br>Turbonomic no longer shows DB Cache Hit Rate values since it is not supported in New Relic MySQL version 8.0 and higher. For more information, see the <a href="#">New Relic documentation</a> . |
| OracleDB | Cache Hit Rate, Transactions, Response Time                                                                                                                                                                                        |
| MongoDB  | Virtual Memory, Connections                                                                                                                                                                                                        |

## Adding a New Relic Target

### Note:

If an application is monitored by New Relic, do not add it as a separate Turbonomic application target.

To add New Relic as a target, specify:

- Account ID

The New Relic Account ID.

- REST API Key

The REST API Key *provided by the New Relic platform*. In the New Relic environment, this API key is called the *User key*. For more information, see [New Relic API Keys](#).

- GraphQL API Key

The GraphQL API Key is identical to the New Relic *User key*, which in Turbonomic is called the REST API Key. You can use the New Relic *User key* value for both the GraphQL API Key and REST API Key fields. For more information, see [New Relic API Keys](#).

- EU Region

If checked, Turbonomic will use the EU API endpoints.

- Collect Virtual Machine Metrics

Whether to collect VM metrics from this target. When **Collect Virtual Machine Metrics** is turned on, Turbonomic uses the VM metrics it collects from this target, instead of the VM data that can optionally be collected from related infrastructure targets.

- Proxy Host (Optional)

The IP of the Proxy Host.

- Proxy Port (Optional)

The port required by the proxy.

- Proxy Username (Optional)

The username required by the proxy.

- Proxy Password (Optional)

The password required by the proxy.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

## Actions

### Note:

The specific actions that Turbonomic recommends can differ, depending on the processes that Turbonomic discovers.

For other application components, Turbonomic can recommend actions based on the resources it can discover for the application. For example, Node.js<sup>®</sup> applications report CPU usage, so Turbonomic can generate vCPU resize actions and display them in the user interface.

Turbonomic recommends actions for the New Relic supply chain as follows.

- Application Component

- Suspend VM

Recommendation only.

- Provision VM

Recommendation only.

## Monitored Resources

### Note:

The exact resources this target monitors can differ based on application type. The following list of metrics per entity includes all resources you might see.

For a given VM, the resources you see depend on how the VM is discovered, and whether the VM provides resources for an application discovered by this target:

- If the VM hosts an application that is discovered through this target, then you will see VM metrics discovered through this target.
- If the VM is discovered through a different target, and it does not host any application discovered through this target, then you will see VM metrics discovered through that different target.
- If the VM is discovered through this target, but it does not host any application discovered through this target, then Turbonomic does not display metrics for the VM.

Turbonomic monitors the following resources for the New Relic supply chain:

- Application Component
  - Virtual CPU  
Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Memory  
Virtual Memory is the measurement of memory utilized by the entity.
  - Transactions  
Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Heap  
Heap is the portion of a VM or container's memory allocated to individual applications.
  - Response Time  
Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).
  - Connection  
Connection is the measurement of Database Server connections utilized by applications.
  - Remaining GC Capacity  
Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).
  - Threads  
Threads is the measurement of thread capacity utilized by applications.
- Database
  - Virtual Memory  
Virtual Memory is the measurement of memory utilized by the entity.
  - Transactions  
Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Database Memory  
Database memory (or DBMem) is the measurement of memory utilized by a Database Server.
  - Connections  
Connection is the measurement of Database Server connections utilized by applications.
  - DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Business Transaction

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Service

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Virtual Machine

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.

## MySQL

To manage MySQL databases, Turbonomic can connect to one or more database servers within a defined scope.

### Prerequisites

- User Permissions are enabled on the MySQL Server. See [Enabling User Permissions on MySQL](#)

### Adding a MySQL Database Target

You can add all matching targets within a given scope.

To add a database server as a target, you specify:

- Target ID

- Name displayed in the Turbonomic UI

- Username

- Username of the account Turbonomic uses to connect to the target.

- Password

- Password of the account Turbonomic uses to connect to the target.

- Scope

- A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

- If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- Port Number

The MySQL remote port. If blank, Turbonomic will use the MySQL default port of 3306.

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Actions

Turbonomic recommends actions for the application server supply chain as follows:

- Database Server

- **Resize** (Recommend, only)

- Database memory (DBMem)

- Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.

- A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.

- When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.

- Virtual Machine

- **Resize**

- When multiple resize actions that relate to the same vCenter VM are accepted (manually or automated), they execute together to avoid multiple restarts for the same VM. For example, a VM has both a vCPU resize action and a vMEM resize action. When selecting both actions and clicking Execute in the UI, the actions combine into the fewest vCenter tasks to minimize the disruption on the VM instead of one set of tasks per action. Only the VMware vCenter integration supports this functionality at this time.

- Resize resource capacity

- Change the capacity of a resource that is allocated for the VM. For example, a resize action might recommend increasing the VMem available to a VM. Before recommending this action, Turbonomic verifies that the VM's cluster can adequately support the new size. If the cluster is highly utilized, Turbonomic will recommend a move action, taking into consideration the capacity of the new cluster and compliance with existing placement policies.

- For hypervisor targets, Turbonomic can resize vCPU by changing the VM's socket or cores per socket count. For details, see [VCPU Scaling Controls](#).

- Resize resource reservation

- Change the amount of a resource that is reserved for a VM. For example, a VM could have an excess amount of memory reserved. That can cause memory congestion on the host — A resize action might recommend reducing the amount reserved, freeing up that resource and reducing congestion

- Resize resource limit

Change the limit that is set on the VM for a resource. For example, a VM could have a memory limit set on it. If the VM is experiencing memory shortage, an action that decreases or removes the limit could improve performance on that VM.

#### – **Resize PU / Resize VP (IBM PowerVM)**

Turbonomic does not support native execution of Resize PU (processing units) and Resize VP (virtual processor) actions. However, actions can be executed using external automation platforms through Action Orchestration workflows such as Action Scripts or Webhooks.

##### - Min/Max Observation Period

Observation period settings only apply to Resize VP actions.

For details, see [Aggressiveness and Observation Periods](#).

##### - Increment Constant for Processing Unit

Change the increment constant value to use for VM PU resize actions. PUs can be a decimal value in a VM setting.

For details, see [Increment Constant for Processing Unit](#).

##### - Increment Constant for Virtual Processors

Change the increment constant value to use for VM VP resize actions. VPs must be an integer value in a VM setting.

For details, see [Increment Constant for Virtual Processors](#).

#### – **Move**

Move a VM due to:

- High resource utilization on VM or host
- Excess IOPS or latency in VStorage
- Workload placement violation
- Underutilized host (move VM before suspending host)

#### – **Reconfigure**

Change a VM's configuration to comply with a policy.

For hypervisor targets, Turbonomic can reconfigure VMs that violate vCPU scaling policies. For details, see [vCPU Scaling Controls](#).

## **Monitored Resources**

Turbonomic monitors the following resources for the application server supply chain:

### • Database Server

#### – Database Memory

Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM.

#### – Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

#### – Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

#### – Connections



Connection is the measurement of Database Server connections utilized by applications.

- DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Virtual Machine

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Storage (VStorage)

Virtual Storage is the measurement storage utilized by the entity.

- Storage Access

Storage Access is the measurement of IOPS utilized by the entity.

- Latency

Latency is the measurement of storage latency utilized by the entity.

## Enabling User Permissions on MySQL Server

Follow the following steps in order to enable appropriate user permissions on a MySQL Server.

1. Edit the MySQL server's configuration file.

You must edit the `.cnf` file on the MySQL server to grant user permissions. Open a secure shell session on the server and edit the file. Depending on the platform your MySQL is running on, you'll find the file at different locations:

- Debian Linux:

`/etc/mysql/my.cnf`

- Red Hat Linux (Fedora or Centos):

`/etc/my.cnf`

- FreeBSD Linux:

You must create the file at `/var/db/mysql/my.cnf`

Open the file in an editor and find the section, `[mysqld]`. Then make the following changes:

- Comment out the line:

```
skip-networking
```

Commenting out this line enables remote connections over TCP/Is.

- Bind your MySQL server address

In the config file, add the line:

```
bind-address=<MySQL_IP_Address>
```

- Enable the collection of Transaction metrics

In the config file, add the line:

```
innodb_monitor_enable = trx_rw_commits, trx_nl_ro_commits, trx_ro_commits,
trx_rollbacks
```

For example, if your MySQL server has the address, 123.45.66.77, then after you have bound the IP address and enabled Transaction metrics, the section of the `.conf` file should appear as follows:

```
[mysqld]
user = mysql
pid-file = /var/run/mysqld/mysqld.pid
socket = /var/run/mysqld/mysqld.sock
port = 3306
basedir = /usr
datadir = /var/lib/mysql
tmpdir = /tmp
language = /usr/share/mysql/English
bind-address = 123.45.66.77

skip-networking

Uncomment the following line for MySQL versions 5.6+
innodb_monitor_enable = trx_rw_commits, trx_nl_ro_commits, trx_ro_commits, trx_rollbacks

....
```

When you are done, save the `.conf` file.

**Note:** Some MySQL installations use multiple configuration files. If a setting you made does not have the desired effect, make sure that a different configuration file is not overwriting the value.

## 2. Enable collection of Response Time metrics.

Execute the following command to log into to the MySQL server:

```
$mysql -u root -p mysql
```

Then execute the following SQL commands:

```
UPDATE performance_schema.setup_instruments SET ENABLED = 'YES' WHERE NAME LIKE 'statement/
sql%';
```

```
UPDATE performance_schema.setup_instruments SET TIMED = 'YES' WHERE NAME LIKE 'statement/
sql%';
```

### **Note:**

If you want these changes to take effect each time you restart the MySQL server, add these statements to a file, and start the server with the `--init-file` option. For example, if you name the file `MyInit.txt`, then start the MySQL server with the following option:

```
--init-file=MyInit.txt
```

## 3. Give your Turbonomic server remote access to the database.

If you are not already logged into the MySQL server, execute the following command:

```
$mysql -u root -p mysql
```

Then execute the following commands:

Assume a user named USER\_NAME with a password PWD\_STRING. Then assume that your Turbonomic has an IP address of 10.10.123.45. The following command grants privileges to that Turbonomic, if it connects with the specified user account:

```
GRANT SELECT ON performance_schema.* TO 'USER_NAME'@'10.10.123.45' IDENTIFIED BY 'PWD_STRING';
GRANT PROCESS ON *.* TO 'USER_NAME'@'10.10.123.45' IDENTIFIED BY 'PWD_STRING';
FLUSH PRIVILEGES;
```

Note that the FLUSH PRIVILEGES command causes MySQL to retain these settings upon restart.

When you're finished running these SQL commands, log out of MySQL.

## Oracle

To connect to an Oracle database, you will:

- Add a Dynamic Performance view to the Oracle database
- Configure a service account on the database that Turbonomic can use to log on
- Find the Service Name and port for the database

### Prerequisites

- User permissions that grant access to Turbonomic through a specific user account. See [Creating a Service User Account in Oracle](#).
- Dynamic Performance View (V\$) must be enabled. See "Adding a Dynamic Performance View", below.
- Access through the firewall to the Oracle database port that you specify for the Turbonomic target connection

### Adding a Dynamic Performance View

In order to collect data from the Oracle database, Turbonomic uses the Dynamic Performance View (referred to as V\$). V\$ is not enabled by default. You must run a script to build the tables and views that are necessary to enable V\$. In some environments only the DBA has privileges to run this script.

To enable V\$:

- Open a secure shell session (ssh) on the database host as a system user or a user with the sysdba role
- In the shell session enter the following commands:

```
sqlplus /nolog
connect /as sysdba
CREATE USER My_Username IDENTIFIED BY My_Password container=all;
GRANT CONNECT TO My_Username container=all;
GRANT sysdba TO My_Username container=all;
```

#### Note:

If security or other practices prohibit assigning SYSDBA to this user, you can use the following command to provide access to all V\$ views:

```
GRANT select any dictionary TO My_Username;
```

This creates a user account named My\_Username with full privileges to access the V\$ Dynamic Performance view.

## Adding an Oracle Database to Turbonomic

You can add an individual database server as a target, or you can add all matching targets within a given scope.

To add a database server as a target, specify:

- Target Name

The target name that will display in the Turbonomic user interface.

- Username/Password

Valid client credentials for the database server. For Turbonomic to execute actions, the account must have administrator privileges. Also, you must have enabled user permissions to this user account, including remote access from the Turbonomic server.

- Scope

A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

**Note:**

All database servers in the scope must share the same service name, credentials, and port. For databases that have a different value for any of these, you must create a separate target using those values.

- Oracle Port

The port that connects to the database. You must open the firewall on the database server to allow access through this port. For further information, see "Finding the Service Name and Port", below.

- Oracle Service Name

The service name for the database that you are connecting to.

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Finding the Service Name and Port

To specify an Oracle target, you must provide the service name and port that you want to connect to. To find the service name for your database:

- Open a secure shell session (ssh) on the database host as a system user or a user with the sysdba role
- In the shell session, enter the command, `lsnrctl status`

Find the line that has the string `PROTOCOL=tcp` and note the port number.

- In the shell session enter the following commands:

```
sqlplus /nolog
connect /as sysdba
SELECT SYS_CONTEXT('userenv', 'db_name') FROM dual;
```

Note the service name that displays as a result of these commands.

## Actions

Turbonomic recommends actions for the application server supply chain as follows:

- Database Server

- **Resize** (Recommend, only)

- Database memory (DBMem)

- Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.

- A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.

- When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.

- Virtual Machine

- **Resize**

- When multiple resize actions that relate to the same vCenter VM are accepted (manually or automated), they execute together to avoid multiple restarts for the same VM. For example, a VM has both a vCPU resize action and a vMEM resize action. When selecting both actions and clicking Execute in the UI, the actions combine into the fewest vCenter tasks to minimize the disruption on the VM instead of one set of tasks per action. Only the VMware vCenter integration supports this functionality at this time.

- Resize resource capacity

- Change the capacity of a resource that is allocated for the VM. For example, a resize action might recommend increasing the VMem available to a VM. Before recommending this action, Turbonomic verifies that the VM's cluster can adequately support the new size. If the cluster is highly utilized, Turbonomic will recommend a move action, taking into consideration the capacity of the new cluster and compliance with existing placement policies.

- For hypervisor targets, Turbonomic can resize vCPU by changing the VM's socket or cores per socket count. For details, see [VCPU Scaling Controls](#).

- Resize resource reservation

- Change the amount of a resource that is reserved for a VM. For example, a VM could have an excess amount of memory reserved. That can cause memory congestion on the host — A resize action might recommend reducing the amount reserved, freeing up that resource and reducing congestion

- Resize resource limit

- Change the limit that is set on the VM for a resource. For example, a VM could have a memory limit set on it. If the VM is experiencing memory shortage, an action that decreases or removes the limit could improve performance on that VM.

- **Resize PU / Resize VP (IBM PowerVM)**

- Turbonomic does not support native execution of Resize PU (processing units) and Resize VP (virtual processor) actions. However, actions can be executed using external automation platforms through Action Orchestration workflows such as Action Scripts or Webhooks.

- Min/Max Observation Period

- Observation period settings only apply to Resize VP actions.

- For details, see [Aggressiveness and Observation Periods](#).

- Increment Constant for Processing Unit

Change the increment constant value to use for VM PU resize actions. PUs can be a decimal value in a VM setting.

For details, see [Increment Constant for Processing Unit](#).

- Increment Constant for Virtual Processors

Change the increment constant value to use for VM VP resize actions. VPs must be an integer value in a VM setting.

For details, see [Increment Constant for Virtual Processors](#).

- **Move**

Move a VM due to:

- High resource utilization on VM or host
- Excess IOPS or latency in VStorage
- Workload placement violation
- Underutilized host (move VM before suspending host)

- **Reconfigure**

Change a VM's configuration to comply with a policy.

For hypervisor targets, Turbonomic can reconfigure VMs that violate vCPU scaling policies. For details, see [vCPU Scaling Controls](#).

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Database Server

- Database Memory

Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM.

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Connections

Connection is the measurement of Database Server connections utilized by applications.

- TransactionLog

The utilization of the server's capacity for storage devoted to transaction logs

- DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Virtual Machine

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Storage (VStorage)  
Virtual Storage is the measurement storage utilized by the entity.
- Storage Access  
Storage Access is the measurement of IOPS utilized by the entity.
- Latency  
Latency is the measurement of storage latency utilized by the entity.

## Creating a Service User Account in Oracle

To collect data from the Oracle database, Turbonomic requires a service account that has privileges to access the V\$ Dynamic Performance view. To create this account:

- Open a secure shell session (ssh) on the database host as a system user or a user with the sysdba role
- In the shell session enter the following commands:

```
sqlplus /nolog
connect /as sysdba
CREATE USER My_Username IDENTIFIED BY My_Password container=all;
GRANT CONNECT TO My_Username container=all;
GRANT sysdba TO My_Username container=all;
```

This creates a user account named My\_Username with full privileges to access the V\$ Dynamic Performance view.

### Note:

The above example uses a fictitious username. To comply with Oracle 12C norms, the username should include a prefix of c###.

Some enterprises don't allow accounts with sysdba access. IBM recommends using sysdba, according to the Oracle documentation. However, you can work with your Oracle DBA staff to provide read access to the following views, which are the ones that Turbonomic needs:

- V\$INSTANCE
- V\$LOG
- V\$LOGFILE
- V\$PARAMETER
- V\$PGASTAT
- V\$RESOURCE\_LIMIT
- V\$SGASTAT
- V\$SYS\_TIME\_MODEL
- V\$SYSMETRIC
- V\$SYSSTAT

## SQL Server

Turbonomic supports the following versions of this target:

SQL Server 2012, 2014, 2016, 2017, 2019, and 2022

### Standalone and Clustered SQL Servers

Turbonomic discovers both standalone and clustered SQL Servers, and represents them as Database Server entities in the supply chain. When you set the scope to a Database Server entity and view the Entity

Information chart, the **Server Configuration** field indicates whether that entity is a standalone server or is part of a cluster.

For clustered environments, Turbonomic represents each SQL Server instance as a Database Server entity in the supply chain, and automatically creates a group for these instances. To see all the auto-created groups, go to **Settings > Groups** and then search for group names starting with `MSSQL:Cluster:`.

Click a group name to set the scope to that group. In the resulting supply chain, the Database Server entity shows the number of SQL Server instances in the cluster, and is stitched to the Virtual Machine entity, which represents the corresponding SQL Server nodes. Click **Database Server** to see a list of instances, and identify which instance is currently active and which ones are idle (redundant). Turbonomic only monitors resources for the active instance, and shows resource metrics when you set the scope to that instance.

## Prerequisites

- A user account with SQL permissions including `Connect SQL` and `View Server State` on the database
- The following services must be running, and set to enabled:
  - Net.Tcp Listener Adapter
  - Net.Tcp Port Sharing Service
- TCP/IP is enabled on the port used for Turbonomic discovery.

## Creating a Service User Account

The user account that Turbonomic uses for its service login must include the following:

- The account must exist in the Security folder within the SQL Server Object Explorer, with the following properties:
  - Enable **SQL Server Authentication**
  - Disable **Enforce password policy**
- The account's security properties must include:
  - Permission to connect to the database through SQL
  - Permission to view the server state

## Adding a SQL Server Target

Specify the following:

- Target Name  
Name displayed in the Turbonomic UI
- Username  
Username for the account. This username must not include the AD domain
- AD Domain  
The Active Directory domain used by Turbonomic in conjunction with the Username for authentication. Leave blank for local accounts.
- Password  
Password for the account. This username must not include the AD domain
- Scope  
A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.



If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- Browsing Service Port

The UDP port for the browsing service that listens for incoming connections to the SQL instances running on the SQL Server. If the browsing service is reachable via the specified port, Turbonomic discovers the SQL instances used by the VM group that you defined as your scope, as well as the listening ports on those SQL instances. The default UDP port is 1434.

If the service is unreachable, or if you did not specify a UDP port, Turbonomic will use the TCP port that you specified in the SQLServer Port field to discover SQL instances. See the next item.

- SQLServer Port

The TCP port for the SQL Server. Turbonomic will use this port if the browsing service is unreachable, or if you did not specify a browsing service port. The default TCP port is 1433.

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Monitored Resources

Turbonomic monitors the following resources:

- Database Server

**Note:**

For clustered environments, Turbonomic only monitors resources for the currently active SQL Server instance, and shows resource metrics when you set the scope to that instance.

- Connections

Connection is the measurement of Database Server connections utilized by applications.

- Database Memory

Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM.

- DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Storage Access

Storage Access is the measurement of IOPS utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- TransactionLog

The utilization of the server's capacity for storage devoted to transaction logs

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Virtual Storage (VStorage)

Virtual Storage is the measurement storage utilized by the entity.

- Virtual Machine

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Storage (VStorage)

Virtual Storage is the measurement storage utilized by the entity.

- Storage Access

Storage Access is the measurement of IOPS utilized by the entity.

## Actions

Turbonomic recommends actions for the application supply chain as follows.

- Applications

Without separate targets to discover Guest OS Processes or Application Servers, Turbonomic does not generate actions on applications. Instead, it generates resize actions on the host VMs. For on-prem environments, if host utilization is high enough on the physical machine running the application VM, Turbonomic can also recommend provisioning a new host.

- Virtual Machines

- Provision additional resources (VMem, VCPU)

- Move Virtual Machine

- Move Virtual Machine Storage

- Reconfigure Storage

- Reconfigure Virtual Machine

- Suspend VM

- Provision VM

## JBoss

Turbonomic supports connecting to JBoss targets running in these operation modes:

- Managed Domain

A collection of JBoss servers in a domain, with a single Domain Controller process acting as the central management control point. In this case, the VM that hosts the Domain Controller will be the target. You configure a Domain Controller via the `domain-controller` entry in the host `.xml` file on the JBoss machine.

- Standalone

A single JBoss server. The VM that hosts the server is the target.

## Prerequisites

- The target VM is properly configured as Domain Controller or standalone, depending on the operation mode.
- Turbonomic discovers JBoss servers that are running on VMs, or on containers that are deployed on VMs. To set the target for a JBoss server, you must have first discovered the hosting VM through a hypervisor target.

For information about hypervisor targets, see [Hypervisor Targets](#).

- Turbonomic monitors the Threads resource in application servers to track utilization of thread pool capacity. To monitor threads in JBoss, each JBoss server must define a thread pool in its configuration files. For instructions on specifying thread pools, see the JBoss documentation.
- JBoss supports Role-Based Access Control (RBAC) as a way to specify a set of permissions for user accounts.

JBoss disables RBAC by default. If you enable RBAC, Turbonomic requires the `Monitor` role for the target JBoss account in order to discover JBoss servers.

## Adding a JBoss Target

You can add an individual JBoss server as a target, or you can add all matching targets within a given scope.

To add a server as a target, specify:

- Target Name

Name displayed in the Turbonomic user interface

- Username

The username for a JBoss account that has a role with the necessary permissions.

- Password

The password for the JBoss account that you will use.

- Scope

A group of JBoss servers, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

After you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group may result in delays in discovery and monitoring. If larger numbers of VMs are to be targeted, the recommendation is to split them across smaller groups in multiple targets.

- JBoss Webservice port

The port for connections to the JBoss server. The default port for HTTP access is 9990.

- Full Validation

When enabled, Turbonomic will require all JBoss servers hosted on the VMs in the selected scope to be valid targets. If Turbonomic is unable to authenticate a JBoss server in the scope, the target will not validate and data will not be collected.

- Secure Connection

When enabled, Turbonomic will connect to JBoss servers via HTTPS. Be sure that the required certificate has been configured for use on the host.

- Proxy Information

Specify proxy information only if you connect to a JBoss instance via proxy.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username to use with the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

## Application Component Names

Turbonomic displays discovered JBoss servers in the user interface. These names indicate whether they're standalone or in a managed domain.

The name is divided into three sections:

- Domain Controller name or “STANDALONE”
- The server name
- The name or IP address of the VM hosting the JBoss server

For example:

STANDALONE:acm-jboss73 [ACM-JBoss7.3-171.47]

## Actions

Turbonomic recommends actions for the application supply chain as follows.

- Application Component
  - Resize Heap  
Recommendation only.
  - Resize Thread Pool  
Recommendation only.
  - Resize Connection Capacity  
Recommendation only.

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Application Component
  - Transactions  
Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Heap  
Heap is the portion of a VM or container's memory allocated to individual applications.
  - Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Threads

Threads is the measurement of thread capacity utilized by applications.

- Connection

Connection is the measurement of Database Server connections utilized by applications.

- Remaining GC Capacity

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

## Apache Tomcat

Turbonomic supports connecting to individual Tomcat targets. Turbonomic connects to the Tomcat process as a remote client via remote JMX access. Target configuration includes the port used by the JMX/RMI registry.

### Prerequisites

- A valid JMX user account for the Tomcat server.

If Tomcat security is enabled, this must be a Tomcat JMX user with a `readonly` role.

- Tomcat should run on JVM version 7 or 8

- For VMware environments, VMware Tools must be installed on the VM that hosts the Tomcat server. For Hyper-V environments, Hyper-V Integration Services must be installed.

This ensures that the VM hosting the Tomcat server can get its IP address.

- Remote JMX access is enabled through a port that is opened to the firewall.
- Discovered infrastructure.

Turbonomic discovers Tomcat servers that are running on VMs or containers. The hosting VM or container must already be in your Turbonomic inventory.

To set the target for a server running on a VM, you must have first discovered the hosting VM through a Hypervisor target. To set the target for a server running in a container, you must have configured container discovery for Tomcat applications.

- For information about container targets, see [Kubernetes Platform Targets](#)
- For information about hypervisor targets, see [Hypervisor Targets](#)

### Configuring JMX Remote Access

Turbonomic monitors and controls the Tomcat server via JMX Remote access. You must configure a JMX Remote port.

Note that to work with a firewall you should also set the RMI Server port — If you don't set an RMI port, then JMX sets an arbitrary *ephemeral port*, and you can't guarantee that the port will be open to your firewall.

There are two ways to set JMX Remote port on Linux platforms:

- Ports specified as system properties

You can set the port via the system property, `com.sun.management.jmxremote.port`. For example:

```
com.sun.management.jmxremote.port=8050
```

A common way to set this property is to declare it in the CATALINA\_OPTS system variable — You can set this in the `setenv.sh` script. For example:

```
CATALINA_OPTS="$CATALINA_OPTS
```

```
-Dcom.sun.management.jmxremote
```

```
-Dcom.sun.management.jmxremote.port=8050"
```

```
export CATALINA_OPTS
```

Note that this sets the JMX Remote port, but it does not set the RMI Server port — Tomcat startup will specify an ephemeral port for the RMI server.

- Ports specified in a JMX Remote Lifecycle Listener

This listener component fixes the ports used by the JMX/RMI Server. When you configure the listener, you specify both the JMX Remote port and the RMI Server port. This is the preferred method when working with a firewall. For more information, see the Apache Tomcat documentation.

On Windows, the typical installation is with Tomcat as a service. There are two ways to set the JMX Remote port:

- Via `setenv.bat`

Add the property to the CATALINA\_OPTS environment variable:

```
set "CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote.port=8050"
```

- Use the Tomcat configuration utility (`tomcat7w` or `tomcat8w`)

Set the port with the following command:

```
-Dcom.sun.management.jmxremote.port=8050"
```

To discover the JMX port that is set to an already running Tomcat, you can look in the following locations:

- For Linux platforms, look in the configuration files — Either:
  - `setenv.sh` — Assuming you configured the port by adding it to the CATALINA\_OPTS environment variable
  - ``${CATALINA_HOME}/conf/server.xml` — Assuming you configured a JMX Remote Lifecycle Listener in this file
- For Windows platforms, look in:
  - `setenv.bat` — Assuming you configured the port by adding it to the CATALINA\_OPTS environment variable
  - The Windows registry — Assuming you installed Tomcat as a Windows service using the Tomcat Configuration utility

## Adding a Tomcat Target

You can add an individual Tomcat server as a target, or you can add all matching servers within a given scope.

To add a server as a target, specify:

- Target Name  
Name displayed in the Turbonomic UI
- Username  
Username of an account with the Admin role

- Password

Password of an account with the Admin role

- Scope

A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- JMX Remote Port

A JMX port that is set to an already running Tomcat process.

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username to use with the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

## Actions

Turbonomic recommends actions for the application supply chain as follows.

- Application Component (Tomcat Application)

- Resize Heap

Recommendation only.

- Resize Thread Pool

Recommendation only.

- Resize Connection Capacity

Recommendation only.

- Virtual Machines

- Provision additional resources (VMem, VCPU)

- Move Virtual Machine

- Move Virtual Machine Storage

- Reconfigure Storage
- Reconfigure Virtual Machine
- Suspend VM
- Provision VM

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Application Component
  - Virtual Memory
 

Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
 

Virtual CPU is the measurement of CPU utilized by the entity.
  - Transactions
 

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
  - Heap
 

Heap is the portion of a VM or container's memory allocated to individual applications.
  - Response Time
 

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).
  - Threads
 

Threads is the measurement of thread capacity utilized by applications.
  - Connection
 

Connection is the measurement of Database Server connections utilized by applications.
  - Remaining GC Capacity
 

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).
- Virtual Machine
  - Virtual Memory
 

Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
 

Virtual CPU is the measurement of CPU utilized by the entity.

## JVM Application

Turbonomic supports connecting to individual JVM Applications as targets. Turbonomic connects to the JVM process as a remote client via remote JMX access. Target configuration includes the port used by the JMX/RMI registry.

### Prerequisites

- A valid JMX user account for the JVM application
  - If JMX security is enabled this must be a JMX user with a `readonly` role
- The application should run on JVM version 6.0 or higher
- For VMware environments, VMware Tools must be installed on the VM that hosts the application



This ensures that the VM hosting the application can get the application's IP address

- Remote JMX access is enabled through a port that is opened to the firewall
- Discovered infrastructure

Turbonomic discovers JVM applications that are running on VMs or containers. The hosting VM or container must already be in your Turbonomic inventory.

To set the target for a server running on a VM, you must have first discovered the hosting VM through a Hypervisor target. To set the target for a server running in a container, you must have configured container discovery for JVM applications.

- For information about container targets, see [Kubernetes Platform Targets](#)
- For information about hypervisor targets, see [Hypervisor Targets](#)

## Configuring JMX Remote Access

Turbonomic monitors and controls JVM applications via JMX Remote access. You must configure a JMX Remote port.

Note that to work with a firewall you should also set the RMI Server port — If you don't set an RMI port, then JMX sets an arbitrary *ephemeral port*, and you can't guarantee that the port will be open to your firewall.

To set the JMX Remote port, pass in the port at the command line when you start your application. For example, to set the port to 8090, start your application with the following options:

```
-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=8090
```

## Adding JVM Application Targets

When you configure JVM targets, you declare a given scope and add all matching applications within that given scope. To do this, specify:

- **Scope:**

A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- **Port Number:** The JMX Remote port
- **Username/Password:** Credentials for a user account with an Admin role

The credentials you provide must match the credentials you specify for JMX login configuration when you start up the application.

If you disable authentication on the application, then you must still provide arbitrary values for **Username** and **Password**. To disable JMX authentication, use the following flags in the command line as you start the application:

```
-Dcom.sun.management.jmxremote.authenticate=false
```

```
-Dcom.sun.management.jmxremote.ssl=false
```

## Multiple JVM Targets On Single VM

Note that you can specify targets with different ports, but that run on the same VM (use the same IP address). You can also specify targets via the same scope, but with different ports — This is another way to assign applications running on the same VM to different ports. To do this:

To do this, add the targets in two separate steps. For example, assume you want to add two JVM application targets, and they both run on the VM at 10.10.123.45. One application is on port 123, and the other application is on port 456. To specify these two targets:

- Specify the first target with the following parameters:

- Scope: VMs\_myCluster.mycorp.com
- Port number: 123
- Username: AppUser
- Password: \*\*\*\*\*

Then click **ADD**.

- Specify the second target with the following parameters:

- Scope: VMs\_myCluster.mycorp.com
- Port number: 456
- Username: OtherAppUser
- Password: \*\*\*\*\*

Then click **ADD**.

## Actions

Turbonomic recommends actions for the application supply chain as follows.

- Application Component (JVM Application)

- Resize Heap  
Recommendation only.
- Resize Thread Pool  
Recommendation only.
- Resize Connection Capacity  
Recommendation only.
- Suspend VM

This action can only be executed by a VM hosted in a vCenter environment. Applications running on other hypervisors will see only a recommendation.

- Provision VM  
Recommendation only.

- Virtual Machines

- Provision additional resources (VMem, VCPU)
- Move Virtual Machine
- Move Virtual Machine Storage
- Reconfigure Storage
- Reconfigure Virtual Machine
- Suspend VM
- Provision VM

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Application Component (JVM Application)
  - Heap
    - Heap is the portion of a VM or container's memory allocated to individual applications.
  - Remaining GC Capacity
    - Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).
    - Data is collected if JVM profiler is enabled.
- Virtual Machine
  - Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.

## Oracle WebLogic

The typical WebLogic deployment is a managed domain with one Administration Server that provides a single point of entry for administration and management of the domain. The domain can include other WebLogic Servers which are the Managed Servers. You set the WebLogic Administration Server as a Turbonomic target.

For a standalone WebLogic deployment, the single server acts as its own Administration Server — You can set the standalone server as a Turbonomic target.

WebLogic deployments can include clusters to distribute workload across multiple WebLogic servers. Turbonomic recommended actions respect the cluster architecture. For example, if you have enabled horizontal scaling for your WebLogic servers, then Turbonomic can recommend provisioning new servers for a given cluster.

## Prerequisites

- A service user account.
  - To execute actions the service account must have an Admin role. For read-only monitoring and analysis, you can set the target with a more restricted role, but then you will have to execute all recommended actions manually, through the WebLogic interface.
- WebLogic requires both the local and remote ends of the connection be resolvable by DNS. The Turbonomic IP address and all WebLogic server IP addresses must be resolvable by your local DNS server.
- Discovered infrastructure
  - Turbonomic discovers WebLogic servers that are running on VMs or containers. The hosting VM or container must already be in your Turbonomic inventory.
  - To set the target for a server running on a VM, you must have first discovered the hosting VM through a Hypervisor target. To set the target for a server running in a container, you must have configured container discovery for WebLogic applications.
    - For information about container targets, see [Kubernetes Platform Targets](#)
    - For information about hypervisor targets, see [Hypervisor Targets](#)

## Finding the T3 Listen Port

To configure a WebLogic target, you need to know the port that the server listens on for administrative communications. Launch the WebLogic Administration Console:

- Navigate to Domain Structure and display the domain you're interested in
- Navigate to **Environment > Servers** and select the Domain Administration Server you're setting as a target

The console displays configuration information for the server, including the T3 listen port.

## Adding a WebLogic Target

You can add one or more WebLogic targets in a given scope, including both standalone servers and domain managers.

To add a this target, specify:

- Target Name  
Name displayed in the Turbonomic UI
- Username  
Username of an account with the Admin role
- Password  
Password of an account with the Admin role

- Scope  
A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- Port Number  
The WebLogic remote port
- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Actions

Turbonomic recommends actions for the application server supply chain as follows:

- Service  
Turbonomic does not recommend actions to perform on the service itself, but it does recommend actions to perform on the application components and hosting VMs. For example, assume a service that manages three SQL databases. If a surge in requests degrades performance across all three databases, then Turbonomic can start a new application component to run another instance of the database application, and bind it to the service. On the other hand, if SQL requests drop off so that the load balancer only forwards requests to two of the databases, Turbonomic can suspend the dormant database and unbind it.
- Application Component

- Resize Heap

This action can only be executed by Turbonomic when running in a domain controller. Standalone applications will see only a recommendation.

- Resize Connection Capacity

Recommendation only.

- Virtual Machines

- Provision additional resources (VMem, VCPU)
- Move Virtual Machine
- Reconfigure Storage
- Suspend VM

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Service

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Application Component

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Heap

Heap is the portion of a VM or container's memory allocated to individual applications.

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Threads

Threads is the measurement of thread capacity utilized by applications.

- Connection

Connection is the measurement of Database Server connections utilized by applications.

- Remaining GC Capacity

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

## IBM WebSphere

The typical WebSphere deployment is a cell of WebSphere servers, controlled by a Deployment Manager. A cell makes up a managed domain that incorporates multiple VMS that host managed application servers. The Deployment Manager is a WebSphere instance that provides a single point of entry for the managed domain.

**Note:**

When adding a WebSphere Deployment Manager as a target, you must ensure that the name of each WebSphere node is resolvable to an IP address by the Turbonomic instance.

You may need to make changes to your DNS or the file `/etc/resolv.conf` on the Turbonomic instance to make it aware of the domain names in use in your environment.

To configure the WebSphere installation, you can use the WebSphere Integrated Solutions Console. This is a client that exposes configuration settings including the SOAP port and the PMI settings.

To manage the servers in an installation, WebSphere uses the Performance Monitoring Infrastructure (PMI). Each WebSphere server runs a PMI service that collects performance data from the various application server components. Turbonomic uses PMI for monitoring and control of the WebSphere installation.

## Prerequisites

- The PMI service set to monitor at the Basic level or greater
- A service user account

To execute actions the service account must have an Administrator role. For read-only monitoring and analysis, you can set the target with a more restricted role (Monitor), but then you will have to execute all recommended actions manually, through the WebSphere interface.

- Discovered infrastructure

Turbonomic discovers WebSphere servers that are running on VMs or containers. The hosting VM or container must already be in your Turbonomic inventory.

To set the target for a server running on a VM, you must have first discovered the hosting VM through a Hypervisor target. To set the target for a server running in a container, you must have configured container discovery for WebSphere applications.

- For information about hypervisor targets, see [Kubernetes Platform Targets](#)
- For information about container targets, see [Hypervisor Targets](#)

## Finding the SOAP Connector Address

To configure a WebSphere target, you need to know the port that the server listens on for administrative communications. Launch the WebSphere Administration Console:

- Navigate to System **Administration > Deployment Manager**
- Under **Additional Properties**, click **Ports**

The entry for `SOAP_CONNECTOR_ADDRESS` gives the currently set port number.

## Adding a WebSphere Target

You can add an individual WebLogic server as a target, or you can add all matching targets within a given scope.

To add a server as a target, specify:

- Target Name  
Name displayed in the Turbonomic UI
- Username  
Username of an account with the Admin role
- Password  
Password of an account with the Admin role
- Scope

A group of applications, stitched to the underlying VMs when the VMs are discovered as part of a separate Turbonomic target.

If you set the target scope, Turbonomic scans each VM within that group or cluster and tries to connect to the target over the specified port. Turbonomic adds any instances of the target it finds as entities from which metrics are retrieved.

The maximum supported size of the group is 500 VMs, and the recommended size is 250 VMs. Adding more VMs to the group can result in poor performance for discovery and monitoring. To target a larger number of VMs by scope, you should split them across smaller groups and set each group as the scope for a separate target.

- Port Number

The WebSphere remote port

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Actions

Turbonomic recommends actions for the application server supply chain as follows:

- Service

Turbonomic does not recommend actions to perform on the service itself, but it does recommend actions to perform on the application components and hosting VMs. For example, assume a service that manages three SQL databases. If a surge in requests degrades performance across all three databases, then Turbonomic can start a new application component to run another instance of the database application, and bind it to the service. On the other hand, if SQL requests drop off so that the load balancer only forwards requests to two of the databases, Turbonomic can suspend the dormant database and unbind it.

- Application Component

- Resize Heap

This action can only be executed by Turbonomic when running in a domain controller. Standalone applications will see only a recommendation.

- Resize Connection Capacity

Recommendation only.

- Virtual Machines

- Provision additional resources (VMem, VCPU)

- Move Virtual Machine

- Reconfigure Storage

- Suspend VM

## Monitored Resources

Turbonomic monitors the following resources for the application server supply chain:

- Service

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Application Component

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.
- Virtual CPU
  - Virtual CPU is the measurement of CPU utilized by the entity.
- Transactions
  - Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
- Heap
  - Heap is the portion of a VM or container’s memory allocated to individual applications.
- Response Time
  - Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).
- Threads
  - Threads is the measurement of thread capacity utilized by applications.
- Connection
  - Connection is the measurement of Database Server connections utilized by applications.
- Remaining GC Capacity
  - Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

## Instana

Turbonomic supports discovery of applications that are managed by the Instana platform. Turbonomic includes the discovered information about these applications in its calculations for environment health.

### Note:

For Kubernetes environments, Turbonomic stitches NewRelic, AppDynamics, Instana, and Dynatrace containerized application components into the supply chain to provide a unified view of your applications. For more information, see [Cloud Native Targets](#).

### Prerequisites

- Instana release-242 or later

Turbonomic version 8.8.3 (or later) requires release-242 or later to manage an on-prem or SaaS instance of Instana.

- An on-prem or SaaS instance of Instana

This instance must be configured to monitor applications running in your environment.

For an on-prem instance, you must also install [BeeInstana](#). If BeeInstana is not installed, Turbonomic will return a `NullPointerException` and will not be able to collect metrics for Application Components and the underlying entities in the supply chain.

- Virtual machines and/or containers that are running applications managed by Instana

For Turbonomic to discover applications through Instana, the applications must be running on virtual machines or in Kubernetes containers located in your environment. Also, these entities must be managed by Turbonomic hypervisor, public cloud, or Kubernetes targets.

- A valid Instana API token

This must be an API token created with the default permissions.



## Instana Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the target and Turbonomic.

| Instana                 | Turbonomic            |
|-------------------------|-----------------------|
| Application             | Business Application  |
| Database                | Database Server       |
| Endpoint                | Business Transaction  |
| Service                 | Service               |
| Process                 | Application Component |
| Docker / Crio Container | Container             |
| Host                    | Virtual Machine       |

## Instana Permissions

| Turbonomic Functionality | Required Permissions                                                                                                   |
|--------------------------|------------------------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"><li>• Default</li></ul> Turbonomic uses the default API Token for monitoring access. |

## Adding an Instana Target

### Note:

It is possible to monitor certain applications or database servers with both Instana and another Turbonomic target. You should avoid such a configuration because it can cause Turbonomic to generate duplicate entities in the supply chain.

If you monitor an application via an Instana server, and you configure that Instana server as a Turbonomic target, then be sure you have not configured a separate application target in Turbonomic for that same application.

To add an Instana server instance as a target, specify:

- Hostname or IP Address

The host name or IP of the Instana server.

- API Token

The Instana API Token.

- Collect Virtual Machine Metrics

Whether to collect VM metrics from this target. When **Collect Virtual Machine Metrics** is turned on, Turbonomic uses the VM metrics it collects from this target, instead of the VM data that can optionally be collected from related infrastructure targets.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the Instana server via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username for the account to log into the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy over HTTPS.

## Actions

Turbonomic does not recommend actions for Instana Business Applications, Business Transactions, or Services. However, it considers resource utilization by these applications when recommending actions for the underlying virtual machines or containers. Turbonomic propagates this underlying information upward in the supply chain to show the impact infrastructure risks can have on the Business Applications, Transactions, and Services.

Turbonomic recommends actions for:

- Application Component

- Resize Heap

- Recommendation only.

- Database Server

- Resize Database Memory (DBMem) - MySQL and SQL Server only

- Recommendation only.

- Actions to resize DBMem are driven by data on the Database Server, which is more accurate than data on the hosting VM.

- Resize Connection Capacity - Oracle only

- Recommendation only.

## Monitored Resources

### Note:

The exact resources monitored will differ based on application type. This list includes all resources you may see.

Turbonomic monitors the following resources for the Instana supply chain:

- Business Application

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Business Transaction

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Service

- Response Time

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Application Component

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.

- Remaining GC Capacity (Java only)

- Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

- Heap (Java only)

- Heap is the portion of a VM or container's memory allocated to individual applications.

- Database Server

- Virtual CPU (Oracle, MySQL, and SQL)

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory (Oracle, MySQL, and SQL)

- Virtual Memory is the measurement of memory utilized by the entity.

- Database Memory (MySQL and SQL)

- Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

- DB Cache Hit Rate (Oracle, MySQL, and SQL)

- DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Response Time (Oracle)

- Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions (Oracle)

- Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Connections (Oracle)

- Connection is the measurement of Database Server connections utilized by applications.

- Container

- Virtual Memory (Docker containers only)

- Virtual Memory is the measurement of memory utilized by the entity.

- Virtual Machine

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

## Cloud Native Targets

Containers support separation of concerns in a way similar to virtual machines, but allow greater flexibility and use far less overhead. Containers can be deployed singly (uncommon) or within a cluster containing multiple nodes. A single container can implement a complete application, or one container can implement a single process that contributes to a larger, distributed application.

To support cloud native environments, Turbonomic targets Kubernetes clusters. Turbonomic supports target clusters managed on Kubernetes v1.8 or higher, whether the clusters are managed directly via kubeadm, or via other platforms including:

- Red Hat OpenShift
- Pivotal Kubernetes Service
- Amazon Elastic Kubernetes Service (EKS)
- Azure Kubernetes Service (AKS)
- Cisco Container Platform (CCP)
- Google Kubernetes Engine (GKE)

With Cloud Native targets, Turbonomic discovers entities related to container platforms in your environment. Discovery can also stitch the container cluster entities together with managed applications. For example, discovery can show the full application stack if your container environment includes applications managed by the following technologies, and you have added them as targets to Turbonomic:

- [Cisco AppDynamics](#)
- [Instana](#)
- [Dynatrace](#)
- [New Relic](#)

### Prerequisites

- Running Kubernetes 1.8+ Cluster

### Adding a Kubernetes Target

Detailed steps for adding a Kubernetes target can be found on the [KubeTurbo Wiki](#).

### Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the targets and Turbonomic.

| Cloud Native Entity | Turbonomic Entity                   |
|---------------------|-------------------------------------|
| Service             | <a href="#">Service</a>             |
| Container           | <a href="#">Container</a>           |
| A Container's Spec  | <a href="#">Container Spec</a>      |
| Pod                 | <a href="#">Container Pod</a>       |
| Controller          | <a href="#">Workload Controller</a> |

| Cloud Native Entity | Turbonomic Entity                 |
|---------------------|-----------------------------------|
| Namespace           | <a href="#">Namespace</a>         |
| Cluster             | <a href="#">Container Cluster</a> |
| Node                | <a href="#">Virtual Machine</a>   |

**Note:**

If a container pod is attached to a [volume](#), Turbonomic discovers it as a Persistent Volume (PV), and shows which pods are connected to the PV.

**Monitored Resources**

Turbonomic monitors the following resources:

- **Service**

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transaction

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- **Container**

- Virtual Memory (VMem)

VMem is the virtual memory utilized by a container against the memory limit. If no limit is set, node capacity is used.

- VMem Request

If applicable, VMem Request is the virtual memory utilized by a container against the memory request.

- VCPU

VCPU is the virtual CPU (in mCores) utilized by a container against the CPU limit. If no limit is set, node capacity is used).

- VCPU Request

If applicable, VCPU Request is the virtual CPU (in mCores) utilized by a container against the CPU request.

- VCPU Throttling

VCPU Throttling is the throttling of container virtual CPU that could impact response time, expressed as the percentage of throttling for all containers associated with a Container Spec. In the Capacity and Usage chart for containers, *used* and *utilization* values reflect the actual throttling percentage, while *capacity* value is always 100%.

- **Container Spec**

Turbonomic monitors the historical usage of any instance of a container running for the workload (assuming the workload name stays the same). Charts show the trend of usage even with restarts or redeployments.

- **Container Pod**

- VMem

VMem is the virtual memory utilized by a pod against the node physical capacity.

- VMem Request

VMem Request is the virtual memory request allocated by a pod against the node allocatable capacity.

- VCPU

VCPU is the virtual CPU (in mCores) utilized by a pod against the node physical capacity.

- VCPU Request

VCPU Request is the virtual CPU request (in mCores) allocated by a pod against the node allocatable capacity.

- VMem Request Quota

If applicable, VMem Request Quota is the amount of virtual memory request a pod has allocated against the namespace quota.

- VCPU Request Quota

If applicable, VCPU Request Quota is the amount of virtual CPU request (in mCores) a pod has allocated against the namespace quota.

- VMem Limit Quota

If applicable, VMem Limit Quota is the amount of virtual memory limit a pod has allocated against the namespace quota.

- VCPU Limit Quota

If applicable, VCPU Limit Quota is the amount of virtual CPU limit (in mCores) a pod has allocated against the namespace quota.

- **Workload Controller**

Turbonomic monitors quotas (limits and requests) for VCPU and VMem.

- **Namespace**

- VMem Request Quota

VMem Request Quota is the total amount of virtual memory request for all pods allocated to the namespace against the namespace quota.

- VCPU Request Quota

VCPU Request Quota is the total amount of virtual CPU request (in mCores) for all pods allocated to the namespace against the namespace quota.

- VMem Limit Quota

VMem Limit Quota is the total amount of virtual memory limit for all pods allocated to the namespace against the namespace quota.

- VCPU Limit Quota

VCPU Limit Quota is the total amount of virtual CPU limit (in mCores) for all pods allocated to the namespace against the namespace quota.

- **Container Cluster**

Turbonomic monitors resources for the containers, pods, nodes (VMs), and volumes in a cluster.

- **Node (VM)**

Turbonomic monitors the following resources for nodes that host Kubernetes pods. These resources are monitored along with the resources from the infrastructure probes, such as vCenter or a public cloud mediation probe.

- VMem

VMem is the virtual memory currently used by all containers on the node. The capacity for this resource is the Node Physical capacity.

- VCPU

VCPU is the virtual CPU currently used by all containers on the node. The capacity for this resource is the Node Physical capacity.

- Memory Request Allocation

Memory Request Allocation is the memory available to the node to support the ResourceQuota request parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- CPU Request Allocation

CPU Request Allocation is the CPU available to the node to support the ResourceQuota request parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- Virtual Memory Request

Virtual Memory Request is the memory currently guaranteed by all containers on the node with a Memory Request. The capacity for this resource is the Node Allocatable capacity, which is the amount of resources available for pods and can be less than the physical capacity.

- Virtual CPU Request

Virtual CPU Request is the CPU currently guaranteed by all containers on the node with a CPU Request. The capacity for this resource is the Node Allocatable capacity, which is the amount of resources available for pods and can be less than the physical capacity.

- Memory Allocation

Memory Allocation is the memory ResourceQuota limit parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- CPU Allocation

CPU Allocation is the CPU ResourceQuota limit parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

## Actions

Turbonomic recommends actions for the Kubernetes container platform supply chain as follows.

- **Service**

- Provision or Suspend**

- For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

- For details, see [Actions for Kubernetes Services](#).

- **Container**

- Resize (via Workload Controllers)**

- Resize containers to assure optimal utilization of resources. By default, containers resize consistently, which allows all replicas of the same container for the same workload type to resize any resource consistently.

- Turbonomic shows and executes container resize actions via Workload Controllers. You will *not* see actions when you set the scope to containers.

- For details, see [Workload Controller Actions](#).

- **Container Spec**

- None**

- A Container Spec retains the historical utilization data of ephemeral containers. Turbonomic uses this data to make accurate container resize decisions, but does not recommend actions for the Container Spec itself.

- **Container Pod**

- **Move**

- Move a pod between nodes (VMs) to address performance issues or improve infrastructure efficiency. For example, if a particular node is congested for CPU, you can move pods to a node with sufficient capacity. If a node is underutilized and is a candidate for suspension, you must first move the pods before you can safely suspend the node.

- **Provision/Suspend**

- For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, provision or suspend pods associated with those Services to maintain SLOs for your applications.

- When recommending node provision or suspend actions, Turbonomic will also recommend provisioning pods (based on demand from DaemonSets) or suspending the related pods.

- For details, see [Container Pod Actions](#).

- **Workload Controller**

- Resize Containers**

- A Workload Controller executes [container](#) actions. When you set the scope to a Workload Controller and view the actions list, the actions apply to containers. Turbonomic does not recommend actions for the Workload Controller itself.

- For details, see [Workload Controller Actions](#).

- **Namespace**

- Resize Quota**

- Turbonomic treats quotas defined in a namespace as constraints when making container resize decisions. If existing container actions would exceed the namespace quotas, Turbonomic recommends actions to resize up the affected namespace quota.

- Note that Turbonomic does not recommend actions to resize *down* a namespace quota. Such an action reduces the capacity that is already allocated to an application. The decision to resize down a namespace quota should include the application owner.

- For details, see [Namespace Actions](#).

- **Container Cluster**

- None**

- Turbonomic does not recommend actions for a Container Cluster. Instead, it recommends actions for the containers, pods, nodes (VMs), and volumes in the cluster. Turbonomic shows all of these actions when you scope to a Container Cluster and view the Pending Actions chart.

- **Node (VM)**

- A Kubernetes node (cloud or on-prem) is represented as a Virtual Machine entity in the supply chain.

- **Provision**

- Provision nodes to address workload congestion or meet application demand.

- **Suspend**

- Suspend nodes after you have consolidated pods or defragmented node resources to improve infrastructure efficiency.

- **Reconfigure**

- Reconfigure nodes that are currently in the NotReady state.

- Note:**

- For nodes in the public cloud, Turbonomic reports the cost savings or investments attached to these actions.



For details, see [Node Actions](#).

## Fabric Targets

---

A fabric target is a service that unites compute, network and storage access into a cohesive system. When you connect Turbonomic to fabric targets, it monitors the performance and resource consumption of your fabric interconnects, IO modules, chassis, and physical machines to assure application performance and utilize resources as efficiently as possible.

Once connected, Turbonomic discovers the blade servers that host the VMs, the chassis and datastores that provide resources to the blade servers, the IO modules and fabric interconnects that provide network resources, and the virtual datastores that provide storage resources to the VMs.

As part of this process, Turbonomic will stitch information from the fabric target and connected hypervisor targets to provide more granular data and information related to the applications and VMs running on the hypervisor-stitched blade servers. Combined with other targets, this information will support a top-down, application-driven approach to managing your environment.

For example:

When Turbonomic discovers that blade servers housed in a particular chassis have been designated as vCenter hosts, the supply chain stitches the blade servers and chassis to the corresponding vCenter data center to establish their relationship. When you set the scope to that data center and view the Health chart, you will see the blade servers in the list of hosts. In addition, when the data center is included in a merge policy (a policy that merges data centers for the purpose of VM placement), the VMs in the blade servers apply the policy, allowing them to move between data centers as necessary.

When you add application server targets, your applications and their individual components and services are discovered, enabling a view of your infrastructure from an individual application service to the physical hardware. Adding public cloud targets also allow for workloads to potentially migrate from your UCS infrastructure to the cloud, based on cost or available resources.

### Supply Chain

Fabric targets add IO Module, Fabric Interconnect, Domain, and Chassis entities to the supply chain. The Chassis entities host physical machines (blade servers) — The physical machines also consume network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network, and also hosts the UCS Manager for UCS Targets. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect.

### Monitored Resources

Turbonomic monitors the following resources for the fabric supply chain:

- Virtual Machine
  - Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
    - Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
    - Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
    - Latency is the measurement of storage latency utilized by the entity.
- Blade

- Net
  - The utilization of data through the Blade's network adapters.
- Treated as a host of the underlying Hypervisor (see below)
  - CPU, Mem, etc.
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters
  - Net
    - The utilization of data through the host's network adapters
  - Swap
    - The utilization of the host's swap space
  - Balloon
    - The utilization of shared memory among VMs running on the host. ESX-only
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- I/O Module
  - NetThroughput
    - Rate of message delivery over a port.
- Switch
  - NetThroughput
    - Rate of message delivery over a port.
  - PortChannel
    - Amalgamation of ports with a shared net throughput and utilization.

## Cisco UCS Manager

The Cisco Unified Computing System (UCS) Manager is a management solution that participates in server, fabric, and storage provisioning, device discovery, inventory, configuration, diagnostics, monitoring, fault detection, auditing, and statistics collection.

Turbonomic supports Cisco UCS Manager 2.2+.

UCS integrates all of these resources in a scalable multi-chassis platform to converge administration onto a single point. Managing these various entities on a network fabric with Turbonomic enables automation at the hardware level, including automated provisioning of hosts.

### Prerequisites

- A service account Turbonomic can use to connect to Cisco UCS Manager

## Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the target and Turbonomic.

| UCS                        | Turbonomic |
|----------------------------|------------|
| Server / Blade / Rack Unit | Host       |
| Chassis                    | Chassis    |
| Datacenter                 | Datacenter |
| IO Module                  | IO Module  |
| Fabric Interconnect        | Switch     |
| Network                    | Network    |

Fabric targets add IO Module, Fabric Interconnect (Switch), and Chassis entities to the supply chain. Hosts consume resources from Chassis entities, and network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network, and also hosts the UCS Manager. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect. Be sure that all the FC, Ether and Physical Ports are properly configured with suitable roles in UCS so that the supply chain is populated accurately without disjoints.

## Adding UCS Targets

To add a UCS target, select the **Fabric and Network** category and choose one of the UCS Fabric options to match the version of UCS you want to manage. Then provide the following information:

- **Address:** The IP address of the Cisco UCS Manager

This gives access to the Fabric Manager that resides on the interconnect.

Turbonomic connects to the Cisco UCS Manager via the HTTP protocol by default.

- **Username/Password:** The credentials of the account Turbonomic will use to connect to Cisco UCS Manager.

specify the IP address and credentials for Cisco UCS Manager. Turbonomic discovers the fabric interfaces associated with that manager.

### Note:

When providing a username, if the account is managed in Active Directory you must include the domain in case-sensitive spelling. For example, MyDomain\john is not the same as mydomain\john. For local user accounts, just provide the username.

- **Port Number:** The port Turbonomic will use to connect to the target.
- **Secure Connection:** When checked, Turbonomic will connect using SSL.

## Supply Chain

Fabric targets add IO Module, Fabric Interconnect (Switch), and Chassis entities to the supply chain. Hosts consume resources from Chassis entities, and network connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network, and also hosts the UCS Manager. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect. Be sure that all the FC, Ether and Physical Ports are properly configured with suitable roles in UCS so that the supply chain is populated accurately without disjoints.

## Monitored Resources

Turbonomic monitors the following commodities of the UCS target:

- Host
  - Energy  
Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).
  - Carbon Footprint  
Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.
  - Memory (Mem)  
The utilization of the host's memory reserved or in use
  - CPU  
The utilization of the host's CPU reserved or in use
  - IO  
The utilization of the host's IO adapters
  - Net  
The utilization of data through the host's network adapters
  - Swap  
The utilization of the host's swap space
  - Balloon  
The utilization of shared memory among VMs running on the host. ESX-only
  - CPU Ready  
The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- Chassis
  - Power  
Electricity being consumed by the Chassis.
  - Cooling  
The percentage of the acceptable temperature range that is utilized by this chassis. As the chassis temperature nears the high or low running temperature limits, this percentage increases.
- I/O Module
  - NetThroughput  
Rate of message delivery over a port.
- Switch
  - NetThroughput  
Rate of message delivery over a port.
  - PortChannel  
Amalgamation of ports with a shared net throughput and utilization.

## Actions

Turbonomic recommends actions for the various entities of the UCS Fabric Network as follows:

- Host

- Start
- Provision
- Suspend
- Chassis
  - Provision
- Switch
  - Add Port to Port Channel
  - Remove Port from Port Channel
  - Add Port

## HPE OneView

HPE OneView is a management solution that streamlines provisioning and lifecycle management across compute, storage, and fabric. Through a unified API, infrastructure can be configured, monitored, updated, and re-purposed.

HPE OneView integrates all of these resources in a scalable multi-enclosure platform to converge administration onto a single point. Managing these various entities on a network fabric with Turbonomic enables automation at the hardware level, including automated provisioning of hosts.

### Prerequisites

- A service account Turbonomic can use to connect to HPE OneView.
- HPE OneView 2.0 and compatible hardware.
- The **Banner Page** option for the user account should be disabled in the HPE OneView user interface.
- You should disable **Require Acknowledgement** for the user account in the HPE OneView user interface.

### Adding HPE OneView Targets

To add a HPE OneView as a target, select the **Fabric** category and choose the HPE OneView radio button. Then provide the following information:

- **Address:**

The IP address of the HPE OneView target

This gives access to the Fabric Manager that resides on the VM.

Turbonomic uses the HTTPS protocol by default. In order to force the HTTP protocol, the Address must be entered in one of two ways. For example, an IP of 8.8.8.8 must be entered as http://8.8.8.8 or by using a specific HTTP port, such as 8.8.8.8:80.

- **Username/Password:**

The credentials of the account Turbonomic will use to connect to the HPE OneView target.

specify the IP address and credentials for HPE OneView. Turbonomic discovers the fabric interfaces associated with that instance.

**Note:**

When providing a username, if the account is managed in Active Directory you must include the domain in case-sensitive spelling. For example, MyDomain@john is not the same as mydomain@john. For local user accounts, just provide the username.

### Supply Chain

Fabric targets add IO Module, Fabric Interconnect (Switch), Domain, and Chassis entities to the supply chain. The Chassis entities host physical machines — The physical machines also consume network

connection commodities from IO Modules. The Fabric Interconnect supplies connectivity to the overall network. The Domain serves as the bottom-level pool of network resource, supplying the Fabric Interconnect.

**Note:**

For HPE OneView targets, the "Fabric Interconnect" entity exists as a false "Switch", and only as a pass-through for network resources. Unlike other fabric targets, such as UCS, there is no physical hardware that serves this function.

## Monitored Resources

Turbonomic monitors the following commodities of the HPE OneView target:

- Virtual Machine
  - Energy  
Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).
  - Carbon Footprint  
Carbon footprint is the measurement of carbon dioxide equivalent (CO2e) emissions for a given entity. Turbonomic measures carbon footprint in grams.
  - Virtual Memory  
Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU  
Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)  
Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access  
Storage Access is the measurement of IOPS utilized by the entity.
  - Latency  
Latency is the measurement of storage latency utilized by the entity.
- Host
  - Energy  
Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).
  - Carbon Footprint  
Carbon footprint is the measurement of carbon dioxide equivalent (CO2e) emissions for a given entity. Turbonomic measures carbon footprint in grams.
  - Memory (Mem)  
The utilization of the host's memory reserved or in use
  - CPU  
The utilization of the host's CPU reserved or in use
  - IO  
The utilization of the host's IO adapters
  - Net  
The utilization of data through the host's network adapters
  - Swap  
The utilization of the host's swap space

- Balloon
  - The utilization of shared memory among VMs running on the host. ESX-only
- CPU Ready
  - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.
- Note:**
  - When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.
- Latency
  - The utilization of latency on the datastore.
- I/O Module
  - NetThroughput
    - Rate of message delivery over a port.
- Switch
  - NetThroughput
    - Rate of message delivery over a port.
  - PortChannel
    - Amalgamation of ports with a shared net throughput and utilization.
- DPod (if Network Flow target is present)
  - Memory (Mem)
    - The utilization of the DPod's memory reserved or in use.
  - CPU
    - The utilization of the DPod's CPU reserved or in use.
  - Storage
    - The utilization of the storage attached to the DPod.
  - Flow
    - The utilization of the network flow capacity utilized by the DPod. This is divided into Flow1 (Low Cost) and Flow2 (Medium Cost) utilization.

## Actions

Turbonomic recommends actions for the various entities of the HPE OneView Fabric Network as follows:

- Virtual Machines
  - Provision additional resources (VMem, VCPU)
  - Move Virtual Machine

- Move Virtual Machine Storage
- Reconfigure Storage
- Reconfigure Virtual Machine
- Suspend VM
- Provision VM
- Host
  - Start
  - Provision
  - Suspend
- Switch
  - Add Port to Port Channel
  - Remove Port from Port Channel
  - Add Port
- DPod (if Network Flow target is present)
  - Provision new DPod

## Guest OS Processes Targets

---

Using WMI (Windows Management Instrumentation) or SNMP, Turbonomic can discover OS processes that are running on your VMs. Analysis creates an Application Component entity for each process.

Turbonomic does not generate actions for these discovered Application Components. For many types of applications, you can configure a target dedicated to that application. You should use Guest OS Process targets for cases where there is no other target you can use for the application. For more information, see [Applications and Databases Targets](#).

## WMI

Turbonomic will discover application and operating system resources using WMI (Windows Management Instrumentation) in your entire environment, or a portion of your environment, based on scope.

### Prerequisites

- Underlying VM host targets added to Turbonomic (for discovery)
- A WMI user account Turbonomic can use to connect to the WMI targets. This account can either be an administrator user, or a non-administrator belonging to certain local user groups. For a full list of requirements, see [Creating a WMI User Account](#)
- WMI enabled on target VMs. For assistance in enabling WMI, see [Enabling WMI](#).

#### Note:

For Hyper-V hosts, you must install Hyper-V Integration Services on the target VMs. For more information, please refer to the following integration services TechNet article:

<https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx>

For VMware hosts, you must install VMware Tools on the target VMs.

### Adding WMI Targets

To add WMI targets, select the **Guest OS Processes > WMI** option on the Target Configuration page and provide the following information:

- Target Name



The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

- Username

The username Turbonomic will use to connect to the WinRM service on the Virtual Machine hosting the application. This should not include the Active Directory domain.

- Password

The password Turbonomic will use to connect to the WinRM service on the Virtual Machine hosting the application.

- Scope

Turbonomic will search for Windows applications on the VMs found within the set scope. This scope can be set to the entire environment, single or multiple clusters, or particular virtual machines. Select this option and choose the scope for application discovery.

**Note:**

A single scope (target) must not contain more than 500 virtual machines. Turbonomic recommends utilization of multiple WMI targets for environments exceeding this limit. Contact Turbonomic Support for assistance in using multiple WMI targets.

- Domain Name

The Active Directory domain used by Turbonomic in conjunction with the Username for authentication. Leave blank for local accounts.

- Secure connection

When selected, Turbonomic will use a secure (SSL) connection.

- NTLM for Domain Accounts

This option is selected by default. When selected, Turbonomic will use the NTLM protocol instead of the Kerberos protocol for domain accounts. NTLM is always used for local accounts.

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

## Enabling Guest Load Metrics

**Note:**

For environments that use WMI or SNMP probes, you must configure the Turbonomic installation to enable guest load metrics. Without this configuration, the WMI and SNMP probes will not display discovered guest load data.

To enable guest load metrics, you must edit the CR file and restart the platform:

1. Open `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml` for editing.
2. Enable guest metrics in the `properties` section of the file:

```
properties:
 repository:
 showGuestLoad: true
```

3. Save your changes to the file.
4. Apply your changes.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

5. To restart the component, delete the repository pod.

Execute the command (where **{POD\_ID}** is the ID that is appended to the pod in the `get pods` listing):

```
kubectl delete pod -n turbonomic repository-{POD_ID}
```

## Supported Actions

### Note:

The specific actions that Turbonomic recommends can differ, depending on the processes that Turbonomic discovers.

For other application components, Turbonomic can recommend actions based on the resources it can discover for the application. For example, Node.js<sup>®</sup> applications report CPU usage, so Turbonomic can generate vCPU resize actions and display them in the user interface.

For each discovered entity within the application supply chain, Turbonomic can execute or recommend certain actions, as outlined below.

- Applications

Without separate targets to discover Guest OS Processes or Application Servers, Turbonomic does not generate actions on applications. Instead, it generates resize actions on the host VMs. For on-prem environments, if host utilization is high enough on the physical machine running the application VM, Turbonomic can also recommend provisioning a new host.

- Virtual Machines

- Provision additional resources (VMem, VCPU)
- Move Virtual Machine
- Move Virtual Machine Storage
- Reconfigure Storage
- Reconfigure Virtual Machine
- Suspend VM
- Provision VM

## Monitored Resources

Turbonomic monitors the following resources for the container supply chain:

- WMI/SNMP Application

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Machine

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Storage (VStorage)  
Virtual Storage is the measurement storage utilized by the entity.
- Storage Access  
Storage Access is the measurement of IOPS utilized by the entity.
- Latency  
Latency is the measurement of storage latency utilized by the entity.

## Creating a WMI User Account

The service account Turbonomic uses to connect to WMI should be an Active Directory domain account. Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account for WMI:

1. Add the account to each of the following local groups:

- WinRMRemoteWMIUsers\_\_ (or Remote Management Users)
- Performance Monitor Users

### Note:

These groups are standard Windows Server 2012 security groups. If you are using an earlier version of Windows Server and do not see these groups, contact IBM Support for assistance.

2. Grant permissions to the account.

In the WMI Management console, grant the `Enable Account` and `Remote Enable` advanced security permissions to the service account:

- Open the WMI Management console (`wmimgmt`).
- Right-click **WMI Control (Local)** and choose **Properties**.
- Go to the **Security** tab and then click **Security** to display the **Security for Root** dialog.
- Click **Advanced**, select the service account, and click **Edit**.
- Confirm that **This namespace and subnamespace** is selected.
- Select **Enable Account** and **Remote Enable** and click **OK**.

## Enabling WMI

### Enabling Discovery

To enable Turbonomic discovery of Windows-based Guest Processes, you must configure and enable WMI with WinRM, usually by using a group policy for AD Domains.

When local credentials are provided to Turbonomic, NTLM is the authentication mechanism. When domain credentials are provided, either NTLM or Kerberos can be specified.

### Note:

If Kerberos is specified, the IP address of the target must be resolvable to a host name using DNS.

While these steps will change slightly between Windows Server versions, the general instructions are:

1. Open an elevated PowerShell prompt on the virtual machine(s) to be discovered as a WMI Target.
2. Enable the WS-Management protocol and set the default configuration for remote management.

Execute `winrm quickconfig`.

3. Enable the WinRM *Negotiate* authentication scheme.

This policy is enabled by default. To enable this policy if it is disabled, execute `winrm set winrm/config/service/auth '{@Negotiate="true"}'`.

4. Set the WinRM AllowUnencrypted property to true if non-SSL connections are preferred.

This property must be set on both the server and the client. Note that setting this value to true does not mean that WMI passes sensitive data in an unencrypted form. It will send only the content of the SOAP messages as plain text:

- Server setting:

```
winrm set winrm/config/service '@{AllowUnencrypted="true"}'
```

- Client setting:

```
winrm set winrm/config/client '@{AllowUnencrypted="true"}'
```

5. Restart the Remote Registry service.

After restarting the Remote Registry service, Turbonomic will discover the WMI targets.

## SSL Connections

To connect using SSL, then a certificate must be assigned to WinRM, and an SSL listening socket enabled. Turbonomic will accept self-signed certificates. Complete details can be found in the official WinRM documentation.

Some example SSL-related commands:

- List certificates on the system:

```
ls cert://localmachine/my
```

- Find Thumbprint for the default certificate:

```
7B56C33F029E7605D5C58E5597D0A077FE1D7F1C CN=winsql-server1.corp.mycorp.com
```

- Enable SSL listener:

```
winrm create winrm/config/listener?Address=*&Transport=HTTPS
@{Hostname="winsql-server1.corp.mycorp.com";CertificateThumbprint="7B56C33F029E7605D5C58E5597D0A077FE1D7F1C";Port="5986"}
```

## SNMP

Turbonomic will discover application and operating system resources using SNMP (Simple Network Management Protocol) on Linux hosts within the specified scope of your environment.

### Prerequisites

- Underlying VM host targets added to Turbonomic (for discovery)
- SNMP service enabled/configured on target VMs to allow access to the OIDs required. The credentials used must allow access to the following OIDs:

- 1.3.6.1.2.1.1.1 - sysDescr
- 1.3.6.1.2.1.25.4.2.1.2 - hrSWRunName
- 1.3.6.1.2.1.25.5.1.1 - hrSWRunPerfEntry
- 1.3.6.1.2.1.25.3.3.1.2 - hrProcessorLoad
- 1.3.6.1.4.1.2021.4.5 - memTotalReal
- 1.3.6.1.4.1.2021.4.6 - memAvailReal
- 1.3.6.1.4.1.2021.4.11 - memTotalFree
- 1.3.6.1.4.1.2021.4.14 - memBuffer
- 1.3.6.1.4.1.2021.4.15 - memCached

**Note:**

For environments that include SNMP targets, Turbonomic can discover incorrect memory values for Linux systems. This can occur for the systems that use the net-snmp package, version 5.7.2-43.el7. You should use versions less than 5.7.2-43.el7, or greater than or equal to 5.7.2-47.el7.

For Hyper-V hosts, you must install Hyper-V Integration Services on the target VMs. For more information, please refer to the following integration services TechNet article:

<https://technet.microsoft.com/en-us/library/dn798297%28v=ws.11%29.aspx>

For VMware hosts, you must install VMware Tools on the target VMs.

## Adding SNMP Targets

To add SNMP targets, select the **Guest OS Processes > SNMP** option on the Target Configuration page and provide the following information:

- Target Name

The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

- Scope

Turbonomic will search for SNMP applications on the VMs found within the set scope. This scope can be set to the entire environment, single or multiple clusters, or particular virtual machines. Select this option and choose the scope for application discovery.

**Note:**

A single scope (target) must not contain more than 500 virtual machines. Turbonomic recommends utilization of multiple SNMP targets for environments exceeding this limit.

- Community

The SNMP v2c community string Turbonomic will use to connect to the SNMP agent on the VM

- Port number

The Port number Turbonomic will use to connect to the SNMP agent on the VM

- Full Validation

When selected, Turbonomic will require all database servers hosted on the VMs in the selected scope to be a valid target. If Turbonomic is unable to authenticate a database server in the scope, the target will not validate and data will not be collected.

- Enable SNMPv3

When checked, Turbonomic will use SNMPv3 to connect to the virtual machines in the selected scope

- SNMPv3 Username/SecurityName

The Username/SecurityName that Turbonomic will use to connect to the virtual machines hosting the application

- SNMPv3 Enable Privacy

When checked, Turbonomic will encrypt using the privacy password to keep the connection private.

- SNMPv3 Authentication Password

The Authentication Password Turbonomic will use to connect to the virtual machines in the selected scope. This allows requests to be authenticated, confirming the sender's identity.

- SNMPv3 Privacy Password

The Privacy Password Turbonomic will use to connect to the virtual machines in the selected scope. For requests to be encrypted, the SNMP manager and the SNMP agent must share knowledge of the privacy password associated with the username.

- SHA-1 For Authentication

When checked, Turbonomic uses SHA-1 Authentication to connect to the virtual machines in the selected scope. When unchecked, Turbonomic uses MD5

- AES For Privacy

When checked, Turbonomic will use AES 128 to connect to the virtual machines in the selected scope. When unchecked, Turbonomic uses DES

**Note:**

Turbonomic does not support AES 192 or 256 for SNMP targets.

## Enabling Guest Load Metrics

**Note:**

For environments that use WMI or SNMP probes, you must configure the Turbonomic installation to enable guest load metrics. Without this configuration, the WMI and SNMP probes will not display discovered guest load data.

To enable guest load metrics, you must edit the CR file and restart the platform:

1. Open `/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml` for editing.
2. Enable guest metrics in the `properties` section of the file:

```
properties:
 repository:
 showGuestLoad: true
```

3. Save your changes to the file.

4. Apply your changes.

Execute the command:

```
kubectl apply -f /opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

5. To restart the component, delete the repository pod.

Execute the command (where `{POD_ID}` is the ID that is appended to the pod in the `get pods` listing):

```
kubectl delete pod -n turbonomic repository-{POD_ID}
```

## Supported Actions

**Note:**

The specific actions that Turbonomic recommends can differ, depending on the processes that Turbonomic discovers.

For other application components, Turbonomic can recommend actions based on the resources it can discover for the application. For example, Node.js<sup>®</sup> applications report CPU usage, so Turbonomic can generate vCPU resize actions and display them in the user interface.

For each discovered entity within the application supply chain, Turbonomic can execute or recommend certain actions, as outlined in the following list.

- Applications

Without separate targets to discover Guest OS Processes or Application Servers, Turbonomic does not generate actions on applications. Instead, it generates resize actions on the host VMs. For on-prem environments, if host utilization is high enough on the physical machine running the application VM, Turbonomic can also recommend provisioning a new host.

- Virtual Machines
  - Provision additional resources (VMem, VCPU)
  - Move Virtual Machine
  - Move Virtual Machine Storage
  - Reconfigure Storage
  - Reconfigure Virtual Machine
  - Suspend VM
  - Provision VM

## Monitored Resources

Turbonomic monitors the following resources for the application supply chain:

- WMI/SNMP Application
  - Virtual Memory
 

Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
 

Virtual CPU is the measurement of CPU utilized by the entity.
- Virtual Machine
  - Virtual Memory
 

Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
 

Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
 

Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
 

Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
 

Latency is the measurement of storage latency utilized by the entity.

## Enabling SNMP

Turbonomic requires that SNMP is enabled and configured in order to discover Guest Processes. While these steps will change slightly between OS versions, this topic gives you the general instructions:

### Enabling SNMP v2

1. Obtain and install the SNMP server package for your Linux distribution and version. This is commonly called `net-snmp` in most package managers. Also install the `net-snmp-utils` package if it is available
2. Configure the SNMP daemon by editing `/etc/snmp/snmpd.conf` or running `snmpconf -i` at the command line in order to verify that:
  - SNMP v2c is enabled
  - A read-only community name has been set. This community name will be used by Turbonomic to communicate with the SNMP agent on the VM

#### Note:

The community name you select must have at least read-only access to all OIDs in the system.

3. Configure the daemon to listen on a public interface. Most default installations only listen on 127.0.0.1
4. Start the SNMP daemon process

These long form and short form sample `snmp.conf` files illustrate the necessary changes.

#### Long form, VACM:

```
First, map the community name "mycommunity1" into a "security name"
sec.name source community
com2sec notConfigUser default mycommunity1
Second, map the security name into a group name:
groupName securityModel securityName
group notConfigGroup v2c notConfigUser
Finally, grant the group read-only access to any UUID.
group context sec.model sec.level prefix read write notif
access notConfigGroup "" any noauth
exact all none none
```

#### Short form, older syntax:

```
rocommunity mycommunity1 default system
```

### Verify Your SNMP Setup

Verify that your SNMP setup is successful by using the following command from a remote machine, replacing the community string and IP address:

```
snmpwalk -Os -v 2c -c COMMUNITY_STRING IP_ADDRESS iso.3.6.1.2.1.1.1
```

If successful, the command will return the kernel version of the machine (similar to the output of `uname -a`)

### Enabling SNMP v3

1. Obtain and install the SNMP server package for your Linux distribution and version.

Most package managers call this package `net-snmp`. To verify your configuration in step 5 below, you should also install the `net-snmp-utils` package on the Turbonomic VM.

2. Stop the SNMP service.
3. Create the SNMPv3 user.

Execute the following command:

```
net-snmp-config --create-snmpv3-user [-ro] [-A authpass] [-X privpass] [-a MD5|SHA] [-x DES|AES] [username]
```

For example, the command:

```
i. net-snmp-create-v3-user -ro -A snmpv3authPass -a SHA -X snmpv3encPass -x AES snmpv3user
```

results in output similar to:

adding the following line to `/var/lib/net-snmp/snmpd.conf`:

```
createUser snmpv3user SHA "snmpv3authPass" AES snmpv3encPass
```

adding the following line to `/etc/snmp/snmpd.conf`:

```
rouser snmpv3user
```

4. Verify your SNMP setup.

To verify the configuration, you can execute the following command on the Turbonomic VM:

```
snmpwalk -Os -v 2c -c COMMUNITY_STRING IP_ADDRESS iso.3.6.1.2.1.1.1
```



## Hyperconverged Targets

---

A hyperconverged target is a service that unites compute, network and storage access into a cohesive system. When you connect Turbonomic to hyperconverged targets, it will monitor the performance and resource consumption of your hyperconverged infrastructure to maintain application performance while utilizing resources as efficiently as possible.

As part of this process, Turbonomic will stitch information from the hyperconverged target to the associated hypervisor and fabric targets, supporting Application Resource Management (ARM) and providing deeper insight into the state of the hardware and information related to the entities in the supply chain. Combined with application server targets, this information will support a top-down, application-driven approach to managing your environment.

### Monitored Resources

Turbonomic monitors the following resources for the hyperconverged supply chain, once stitched to your hypervisor and other associated targets:

**Note:** The entities visible in the supply chain depend on what supplemental targets have been added in addition to the hyperconverged target.

- Virtual Machine
  - Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
    - Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
    - Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
    - Latency is the measurement of storage latency utilized by the entity.
- Blade
  - Net
    - The utilization of data through the Blade's network adapters.
  - Treated as a host of the underlying Hypervisor (see below)
    - CPU, Mem, etc.
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters
  - Net
    - The utilization of data through the host's network adapters
  - Swap
    - The utilization of the host's swap space

- Balloon
  - The utilization of shared memory among VMs running on the host. ESX-only
- CPU Ready
  - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- I/O Module
  - NetThroughput
    - Rate of message delivery over a port.
- Switch
  - NetThroughput
    - Rate of message delivery over a port.
  - PortChannel
    - Amalgamation of ports with a shared net throughput and utilization.
- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

  - Latency
    - The utilization of latency on the datastore.
- Disk Array
  - Storage Amount
    - The utilization of the Disk Array's capacity.
  - Storage Provisioned
    - The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the disk array.
  - Latency
    - The utilization of latency, computed from the latency of each device in the disk array.

## Cisco HyperFlex

Cisco HyperFlex provides a hyperconverged platform that combines the networking and compute power of UCS with the storage capabilities of the HyperFlex HX Data Platform.

With the additional and refined storage information provided by HyperFlex, Turbonomic narrows the Desired State and recommends actions using the joint compute and storage information, gaining valuable insight into the interconnected nature of your environment.

For Turbonomic to make the most informed decisions, you must also add the compute nodes and any associated hypervisors. HyperFlex environments typically include:

- Converged (HX) Nodes

A combination of the cluster's storage devices into a single multi-tiered, object-based datastore.

- Compute Nodes

Cisco B or C series servers that make up the compute resources of the cluster, and are typically managed by a hypervisor.

- Controller VMs

Each HyperFlex node includes a Controller VM that intercepts and handles all the I/O from associated virtual machines. Turbonomic will not recommend actions for these VMs.

## Prerequisites

- A service account on the controller VM that Turbonomic can use to connect to the HyperFlex environment

## Adding HyperFlex Targets

To add a HyperFlex target, select the **Hyperconverged > HyperFlex** option on the Target Configuration page and provide the following information:

- Address

The name or IP address of the HyperFlex controller VM. If a port is appended to the IP address, it must be a secure (HTTPS) port.

- Username/Password

Credentials for a user account Turbonomic will use to connect to the controller VM.

To specify a root user for the target, give the name as `local/root`. Note that earlier versions of Turbonomic accepted a username of `root`. If an existing HyperFlex target fails to validate after you have upgraded Turbonomic, then check to make sure the username for a root account is properly set to `local/root`.

If you use vSphere SSO for authentication to the HyperFlex target, you must prepend your username with `vc-`. For example, if your username is `Administrator@vsphere.local`, you would enter it as `vc-Administrator@vsphere.local`.

- Port Number

Port Number Turbonomic will use to connect to the controller VM.

## Supply Chain

HyperFlex targets add Disk Array entities to the supply chain, and receive more granular information from the compute resources in your environment.

## Entity Comparison

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in HyperFlex to those used in Turbonomic:

| HyperFlex Name | Turbonomic Entity |
|----------------|-------------------|
| Volume         | Storage           |
| HX Cluster     | Disk Array        |

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

For each discovered entity, Turbonomic can recommend certain actions, as outlined below.

| Entity Type | Can Be Automated | Recommendations only          |
|-------------|------------------|-------------------------------|
| Storage     |                  | Move, Provision, Resize Up    |
| Disk Array  |                  | Provision, Suspend, Resize Up |

### Note:

For this target, Turbonomic discovers the HX Cluster as a Disk Array. When you see a provision action on this entity, you should determine which of the following is most relevant, based on your environment:

- Add disks to converged nodes
- Add a new converged node
- Add a new HX Cluster

## Monitored Resources

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount  
The utilization of the datastore's capacity.
  - Storage Provisioned  
The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the datastore.

### Note:

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency  
The utilization of latency on the datastore.
- Disk Array
  - Storage Amount  
The utilization of the Disk Array's capacity.
  - Storage Provisioned  
The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the disk array.
  - Latency  
The utilization of latency, computed from the latency of each device in the disk array.

## Nutanix

Nutanix products provide hyperconverged platforms that include VM hosting and a distributed storage fabric. The platform presents storage in two tiers — Local HDD storage and server-attached flash (hot storage).

Nutanix environments may include:

- One or more Nutanix appliances

An appliance contains up to four server nodes.

- Nutanix nodes

Servers that expose compute and storage resources. Each node provides local HDD and hot storage. Nodes combine to form a unified cluster that pools resources.

- Controller VMs

Each node includes a Controller VM that manages the node's resources within the cluster pool. To minimize storage latency, the Controller VM keeps the most frequently accessed data in the hot storage.

Turbonomic supports management of Nutanix fabrics, where the supply chain treats a Nutanix Storage Pool as a disk array. Turbonomic recognizes Nutanix storage tiers when calculating placement of VMs and VStorage. In addition, Turbonomic can recommend actions to scale flash capacity up or down by adding more hosts to the cluster, or more flash drives to the hosts.

To specify a Nutanix target, provide the Cluster External IP address. This is a logical IP address that always connects to one of the active Controller VMs in the cluster. In this way, you can specify a Nutanix target without having to specify an explicit Controller VM.

### Note:

The Controller VM must remain *pinned* to its host machine — You must not move the Controller VM to a different host. If the Nutanix cluster uses the Nutanix Acropolis OS to manage VMs, Turbonomic automatically pins the Controller VMs. However, if you use vCenter Server or Hyper-V to manage VMs on the hosts, you must configure a group to pin the Controller VMs. For more information, see [Pinning Nutanix Controller VMs](#).

## Prerequisites

- A service account with cluster administrator rights on the Nutanix cluster(s) for action execution. For entity discovery, a minimum of READ access is required.

## Finding the Cluster External IP Address

To configure a Nutanix target, provide the Cluster External IP address for the given Nutanix cluster.

The Cluster External IP address is a logical IP that resolves to the cluster's Prism Element Leader. If the Prism Element Leader fails, then the Cluster External IP address will resolve to the newly elected Prism Element Leader.

To find this IP address, open the Web Console (the Prism Element) on the cluster and navigate to the **Cluster Details** view. In this view you can see the **Cluster External IP** address. If there is no IP address specified, you can specify the address at this time. For more information, see the Nutanix documentation.

## Operating Modes

A Nutanix node is a server that hosts VMs — In this sense the node functions as a hypervisor. A cluster of nodes can host VMs using the following Hypervisor technologies:

- Nutanix Acropolis

The native Nutanix host platform, which combines software-defined storage with built-in virtualization.

- VMware ESXi

- Microsoft Hyper-V

## Controller VM Pinning

Each Nutanix node hosts a Controller VM that runs the Nutanix software and manages I/O for the hypervisor and all VMs running on the host. Each Controller VM must remain on its host node —The Controller VM must be *pinned* to that host, and must not be moved to any other host.

For more information about how to pin the Controller VM, see [Pinning Nutanix Controller VMs](#).

## Adding Nutanix Targets

### Note:

This describes how to add a Nutanix cluster to Turbonomic as a target. If Nutanix is not managing hosts running Acropolis as the hypervisor, you will have to add the vCenter or Hyper-V hypervisors as targets after you have added the Nutanix cluster as a target. For more information, see [Hypervisor Targets](#).

To add Nutanix targets, select the **Hyperconverged > Nutanix** option on the Target Configuration page and provide the following information:

- Address  
The Cluster External IP address for the Nutanix cluster.
- Port Number  
The listening port of the cluster.
- Secure Connection  
When Turbonomic will use a secure connection.
- Username/Password  
Credentials for an account on the Nutanix cluster with sufficient privileges.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in Nutanix to those used in Turbonomic:

| Nutanix Name    | Turbonomic Entity  |
|-----------------|--------------------|
| Container       | Storage            |
| Storage Pool    | Disk Array         |
| Nutanix Cluster | Storage Controller |

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type           | Can Be Automated                                                                                                                                                                                                                                       | Recommendations only |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|
| VM (a Nutanix VM)     | Move (Host), Resize<br>Resize actions require the VM to power down, and power back on again.<br><b>Note:</b> Turbonomic can automate VMotion to hosts, but for storage moves on Nutanix Turbonomic only supports the Recommend action acceptance mode. |                      |
| Datastore ("Storage") | Provision, Resize Up, Resize Down, Suspend                                                                                                                                                                                                             | Move                 |
| Disk Array            |                                                                                                                                                                                                                                                        |                      |
| Storage Controller    |                                                                                                                                                                                                                                                        | Provision            |

## Monitored Resources

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount

The utilization of the datastore's capacity.
  - Storage Provisioned

The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.
  - Latency

The utilization of latency on the datastore.
- Disk Array
  - Storage Amount

The utilization of the Disk Array's capacity.
  - Storage Provisioned

The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the disk array.
  - Latency

The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller

**Note:**

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- CPU

- The utilization of the Storage Controller's allocated CPU.

- Storage Amount

- The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## Pinning Nutanix Controller VMs

Each Nutanix node hosts a Controller VM that runs the Nutanix software and manages I/O for the hypervisor and all VMs running on the host. Each Controller VM must remain on its host node —The Controller VM must be *pinned* to that host, and must not be moved to any other host.

For a cluster using vCenter or Hyper-V hypervisors, you must use Turbonomic policies to pin the Controller VMs to their respective nodes. To do this, you will create a dynamic group of Nutanix Controller VMs, and then disable move actions for all members of this group.

To pin the Controller VMs:

1. Create a group of Controller VMs.

- In Turbonomic you can create dynamic groups based on VM name — All VMs with matching names automatically belong to the group. Nutanix uses the following naming convention for Control VMs:

- NTNX-`<SerialNumber>`-A-CVM, where `<SerialNumber>` is the serial number of the Controller VM.

- You can create a dynamic group that automatically includes these Nutanix controller VMs. (For complete instructions on creating groups, see [Creating Groups](#).)

- Create a new group.

- In Turbonomic navigate to **Settings > Groups** and create a new group.

- Set the group type to **Dynamic**.

- Add a filter to match VMs by their names.

- Add a filter that uses the regular expression, `NTNX.*CVM`. This regular expression will match the Nutanix Controller VMs.

- Be sure to save the group. All the Nutanix Controller VMs will automatically become members of this group.

2. Disable moves for all VMs in this group.

- To do this, create an automation policy for the group and disable actions. (For complete instructions to create these policies, see [Creating Automation Policies](#)).

- In Turbonomic go to the **Policy > Action > VM** view.

- Set the scope to the group you made.

- In the **Scope** column, expand **My Groups** and select the group you just made.

- Disable moves for this group.

- In the **Parameter** column under **Action Acceptance**, set the value to **Disabled**. This will override the global action mode.

- Save the action acceptance mode.

- Be sure to click **Apply Settings Change**.



## Hypervisor Targets

---

A hypervisor is a service that creates and runs virtual machines (VMs) and/or containers, providing these entities compute and storage resources. When you connect Turbonomic to hypervisor targets in your environment, Turbonomic assures application performance by utilizing these resources as efficiently as possible.

Once connected to a hypervisor target, Turbonomic discovers the VMs, containers, physical machines that host the VMs or containers, datastores that provide storage resources to the physical machines, and virtual datastores that provide storage resources.

As additional targets are added, Turbonomic will discover the resources belonging to your physical and virtual infrastructure. For example, adding the underlying hardware as part of a UCS and/or storage target will provide additional visibility into the physical infrastructure of your environment. To extend the virtual infrastructure, application server or guest operating process targets can be added.

Turbonomic represents your environment holistically as a supply chain of resource buyers and sellers, all working together to meet application demand. By empowering buyers (VMs, instances, containers, and services) with a budget to seek the resources that applications need to perform, and sellers to price their available resources (CPU, memory, storage, network) based on utilization in real-time, Turbonomic maintains your environment within the desired state.

For more information, see [Application Resource Management](#).

### Supply Chain

Each hypervisor requires a physical machine (host) and one or more datastores to provide compute and storage resources. Virtual machines (VMs) or containers run on those physical resources, and the VMs in turn provide resources to applications.

At the bottom of the supply chain, physical machines consume resources from data centers.

If your environment includes SAN technologies such as disk arrays, then the storage consumes resources from that underlying technology. If you add these storage targets, then Turbonomic extends the supply chain analysis into the components that make up the disk array. For more information, see [Storage Manager Targets](#).

### Actions

Turbonomic recommends actions for the hypervisor supply chain as follows.

**Note:**

This is a general list of actions for entities discovered for hypervisors. Detailed actions per target are described in each target section.

- Virtual Machines
  - Provision additional resources (VMem, VCPU)
  - Move Virtual Machine
  - Move Virtual Machine Storage
  - Reconfigure Storage
  - Reconfigure Virtual Machine
  - Suspend VM
  - Provision VM
- Host
  - Start
  - Provision
  - Suspend

- Storage
  - Start Storage
  - Provision Storage
  - Suspend Storage
  - Move (only with Storage Targets configured)
  - Resize (only with Storage Targets configured)
- Consumer Virtual Datacenters
  - Resize Consumer vDC
  - Provision Consumer vDC

## Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

- Virtual Machine
  - Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
    - Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
    - Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
    - Latency is the measurement of storage latency utilized by the entity.
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters
  - Net
    - The utilization of data through the host's network adapters
  - Swap
    - The utilization of the host's swap space
  - Balloon
    - The utilization of shared memory among VMs running on the host. ESX-only
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- Storage
  - Storage Amount

- The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.
  - Note:**
    - When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.
  - Latency
    - The utilization of latency on the datastore.
- Datacenter
  - Note:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the hosts in the datacenter.
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use.
  - CPU
    - The utilization of the host's CPU reserved or in use.
  - IO
    - The utilization of the host's IO adapters.
  - Net
    - The utilization of data through the host's network adapters.
  - Swap
    - The utilization of the host's swap space.
  - Balloon
    - The utilization of shared of memory among VMs running on the host. ESX-only.
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only .
- Provider Virtual Datacenter
  - Memory (Mem)
    - The utilization of the Datacenter's memory reserved or in use.
  - CPU
    - The utilization of the Datacenter's CPU reserved or in use.
  - Storage
    - The utilization of the storage attached to the Provider vDC.
- Consumer Virtual Datacenter
  - Memory (Mem)
    - The utilization of the Datacenter's memory reserved or in use.
  - CPU
    - The utilization of the Datacenter's CPU reserved or in use.
  - Storage

The utilization of the storage attached to the Consumer vDC.

## IBM PowerVM

For IBM® Power 8, 9, and 10 environments using Hardware Management Console (HMC) versions 9 or 10, Turbonomic will discover and continuously decide on the optimal logical partition (LPAR) Processing Unit (PU) allocation and Virtual Processor (VP) capacity based on historical demand collected from the HMC.

IBM Power Systems and LPARs are represented as Hosts and Virtual Machines and are visible across all HMC targets in the Turbonomic UI. Recommended LPAR resize PU and VP actions indicate optimal LPAR size, and can leverage existing Turbonomic workflow capabilities for external approval and execution.

Turbonomic automatically creates dynamic groups and Automation policies for the IBM Power Systems and LPARs.

- Turbonomic creates two dynamic groups, one for LPARs and one for systems. These groups are scoped to all LPARs and all Systems across all PowerVM targets that are added to the system.

These groups are dynamic – If you allocate or remove entities (VMs, hosts, and so on) , Turbonomic automatically adds or removes those entities in the given group.

The name of each group begins with IBM PowerVM. For example, IBM PowerVM: All LPARs and IBM PowerVM: All Systems.

- Turbonomic creates Automation policies to specify utilization and scaling constraints known to be appropriate for PowerVM environments. The name for each policy begins with the name IBM PowerVM. For more information, see [Automation Policies](#).

Turbonomic then applies these policies to the groups that correspond to the entity type.

## Prerequisites

- Supported Hardware Management Console (HMC) versions:
  - HMC 10 - Power 8/9/10
  - HMC 9 - Power 8/9 only; Power 7 systems are ignored
- Compute-Long Term Monitoring Metrics (LTM) must be enabled on each Power System in HMC.
  - Manually enable Compute-LTM using HMC for each Power System you would like Turbonomic to discover.
  - In Turbonomic, select **Automatically enable HMC Long Term Metric collection on Power Systems** in the target addition and edit pages to enable Compute-LTM on all hosts during discovery. This option is unselected by default.

- An HMC user account that Turbonomic can use to connect to the PowerVM target.

The HMC user account should have the following roles:

- The `hmcviewer` role is required.
  - The `Manage Partition Utilization Data` task role is required to automatically enable Compute-LTM on each HMC Power System without having to do this manually.
- The HTTPS tcp port 443 from Turbonomic to Power HMC instance must allow connections from Turbonomic to the HMC.

## Considerations

Before you configure an IBM PowerVM target, you should consider the following:

- Turbonomic recommends Resize LPAR actions only.
- Stitching to entities above the LPAR/VM or below the System/Host is not currently enabled.
- Where multiple HMCs are managing the same Power Systems (Clustered/HA configuration) only one of the HMC nodes should be added as a target to Turbonomic.

- Plans are not currently supported. Planning features are being considered in a future release.
- Processor pools are not currently supported.
- Reporting using ThoughtSpot is not currently supported.
- The **Details** tab requires manual configuration for static group scopes.
- VIOS LPARs are not discovered at this time.
- In a small number of cases, actions to resize LPAR PU and VP resources may result in recommendations for PU allocations above VP capacity and vice versa.

## Adding IBM PowerVM Targets

To add IBM PowerVM targets, select the **Hypervisors > IBM PowerVM** option on the Target Configuration page and provide the following information:

### Note:

If you do not see the IBM PowerVM target, you must enable the PowerVM probe before adding the target. To enable the probe, set the following in the `charts_v1alpha1_xl_cr.yaml` resource:

```
spec:
 powervm:
 enabled: true
```

- Display name

The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

The name provided here is also used for the Data Center and Cluster name to which Systems managed by this HMC belong.

- Address

The FQDN or IP address of the PowerVM HMC.

- Port number

The port number for the remote management connection. The default HTTPS port is 443.

- Username

The username for a dedicated Turbonomic user account on the PowerVM HMC instance.

- Password

Password for given user account.

- Automatically enable HMC Long Term Metric collection on Power Systems

When selected, Turbonomic automatically enables Compute-Long Term Monitor metrics (LTM) on each power system managed by the HMC. By default, Turbonomic will not make any changes on HMC; however, Compute-LTM must be enabled on each Power System either automatically (enable this option) or manually.

To do this, the user account used by Turbonomic must have `Manage Partition Utilization Data` as a Task Role in HMC.

## Entity Mapping

After validating your targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between the target and Turbonomic.

| HMC                      | Turbonomic           |
|--------------------------|----------------------|
| Logical Partition (LPAR) | Virtual Machine (VM) |

|            |                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HMC</b> | <b>Turbonomic</b>                                                                                                                                         |
| System     | Host                                                                                                                                                      |
| -          | Data Center; skeleton entity used to complete the topology<br>This entity uses the target Display Name. There is one Data Center for each PowerVM target. |

The following table describes the mapping between HMC systems states and Turbonomic.

| <b>HMC System State</b> | <b>Turbonomic Power State</b> |
|-------------------------|-------------------------------|
| INITIALIZING            | UNKNOWN                       |
| OPERATING               | ACTIVE                        |
| STANDBY                 | UNKNOWN                       |
| POWER_OFF               | IDLE                          |

## Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

- Virtual Machine

- Processing Units

For PowerVM, processing units are a unit of measure for shared processing power across one or more virtual processors. One shared processing unit on one virtual processor accomplishes approximately the same work as one dedicated processor.

PUs can be a decimal value in a VM setting. A VM that has a capacity of 0.5 PU means that 0.5 PU is always readily available for that VM to use, and that amount can only be used for that VM.

- Virtual Processors

For PowerVM, virtual processors (VPs) are the guest OS representation of the capacity of the VM. VPs have a one to one correlation with PUs, meaning 1 VP being utilized in a VM equates to 1 PU being used; however, VPs do not have to equal to PUs on a system. VPs are a virtual representation, therefore they can be over-provisioned.

For example, a VM can have 0.5 PUs but 3 VPs. In this scenario a VM's capacity is 0.5 PU but it can use more than its assigned capacity; the max it can use is the number of VPs. The number of VPs must never exceed 20x the value of PU, therefore the highest utilization a VM can be at is 2000% of its PU capacity.

**Note:**

Only shared-uncapped VM utilization can go up to 2000% of its PU capacity; the utilization for all other VM types cannot go above 100% of its PU capacity.

- Host

- Memory (Mem)

The memory given to a VM from the host system; what is provided by the host is all that the VM can use.

- Data Center

**Note:**

The Data Center is not monitored but is used in the supply chain to represent each PowerVM HMC and the Hosts and VMs associated with it.

## Actions

When Turbonomic generates an action, the action details include a list of policies that constrain the action recommendation. In this list, policy names that begin with IBM PowerVM: indicate policies that implement the constraints suitable for PowerVM. You can click to navigate to the policy definition. Note that you can see the policy definition, but you cannot edit it.

The policies that Turbonomic generates affect Turbonomic actions in the following ways:

| Entity          | Policy Impact on Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual Machine | <ul style="list-style-type: none"> <li>• <b>Resize PU / Resize VP (IBM PowerVM)</b></li> </ul> <p>Turbonomic does not support native execution of Resize PU (processing units) and Resize VP (virtual processor) actions. However, actions can be executed using external automation platforms through Action Orchestration workflows such as Action Scripts or Webhooks.</p> <ul style="list-style-type: none"> <li>– Min/Max Observation Period</li> </ul> <p style="padding-left: 20px;">Observation period settings only apply to Resize VP actions.</p> <p style="padding-left: 20px;">For details, see <a href="#">Aggressiveness and Observation Periods</a>.</p> <ul style="list-style-type: none"> <li>– Increment Constant for Processing Unit</li> </ul> <p style="padding-left: 20px;">Change the increment constant value to use for VM PU resize actions. PUs can be a decimal value in a VM setting.</p> <p style="padding-left: 20px;">For details, see <a href="#">Increment Constant for Processing Unit</a>.</p> <ul style="list-style-type: none"> <li>– Increment Constant for Virtual Processors</li> </ul> <p style="padding-left: 20px;">Change the increment constant value to use for VM VP resize actions. VPs must be an integer value in a VM setting.</p> <p style="padding-left: 20px;">For details, see <a href="#">Increment Constant for Virtual Processors</a>.</p> |
| Host            | <ul style="list-style-type: none"> <li>• <b>Memory</b></li> </ul> <p>The memory given to a VM from the host system; what is provided by the host is all that the VM can use.</p> <ul style="list-style-type: none"> <li>– Memory Overprovisioned Percentage</li> </ul> <p style="padding-left: 20px;">Change the percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.</p> <p style="padding-left: 20px;">For details, see <a href="#">Utilization Constraints</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

## Microsoft Hyper-V

If you have a small number of Hyper-V hosts in your environment, you can add them individually as Turbonomic targets. Also, if you have deployed the Hyper-V hosts in a clustered domain (for example as a failover cluster), you can specify one Hyper-V host as a target and Turbonomic automatically add the other members of that cluster.

Note that for large Hyper-V environments, it's typical to manage the hosts via System Center Virtual Machine Manager (VMM). You can specify the VMM server as a target and Turbonomic will use it to discover and manage its child Hyper-V hosts. If you use VMM, you should not add individual Hyper-V hosts as targets. For information about adding VMM targets, see [Adding Virtual Machine ManagerTargets](#).

### Note:

Even if you manage Hyper-V using a VMM target, you must still configure remote management on each Hyper-V server. This Hyper-V topic includes instructions to configure remote management – see [Enabling Windows Remote Management](#).

For accurate SMB storage calculations, Turbonomic requires a VMM target.

## Prerequisites

- Create a user account that Turbonomic can use to connect to your Hyper-V servers. See [Creating a Service User Account in Hyper-V](#)
- Configure remote management on each Hyper-V server. Refer to [Enabling Windows Remote Management](#)
- The time on each Hyper-V host must be in synch with the rest of the managed Hyper-V environment.
- Your Hyper-V environment must not use Server Message Block (SMB) storage.

To manage SMB storage, Turbonomic requires a VMM target, and that VMM instance must manage the Hyper-V hypervisors and the SMB storage that they use.

Managing a Hyper-V plus SMB environment via Hyper-V targets will result in incorrect data collection for SMB storage.

## Adding Hyper-V Targets

Once you've enabled remote management, you can add your Hyper-V hosts as targets. To add Hyper-V targets, select the **Hypervisors > Hyper-V** option on the Target Configuration page and provide the following information:

- Address

The FQDN of the Hyper-V host. If you're using the "Discover Host Cluster" below to add an entire cluster, enter the name of any one of the Hyper-V hosts in the cluster.

Note that you can enter an IP address for the host, but you must first configure an SPN on the host. IBM recommends that you use the FQDN in this field.

- Port number

The port number for the remote management connection. The default HTTP port is 5985; the default HTTPS port is 5986.

- Secure connection

Select this option to use a secure connection (HTTPS). Make sure the required certificate is configured for use on the host.

- Full domain name

The full domain name of the cluster to which the host belongs.

- Discover Host Cluster

Turbonomic discovers and adds all Hyper-V hosts in the named cluster if this option is checked. Note that each server must be configured to allow remote management. You may find it helpful to configure WinRM using a GPO so new servers are configured automatically (see [Enabling WinRM Via a GPO](#)).

- Username

The username of a user account Turbonomic can use to connect to the Hyper-V host. If you checked "Discover Host Cluster" in the field above, use an account that is valid for all Hyper-V hosts in that cluster.

- Password

Password for account used.

### Note:



If your Hyper-V hosts are running in a Nutanix environment, you must understand pinning a Nutanix Controller VM. For more information, see [Pinning Controller VMs in Generic Hypervisor Mode](#).

## Exporting Hyper-V Virtual Machines

In Hyper-V environments, you must be sure that all VMs have unique IDs.

Hyper-V supports the export of a VM, so that you can create exact copies of it by importing those exported files. The Copy import type creates a new unique ID for the imported VM. When importing VMs in your environment, you should always use the Copy import type.

Turbonomic uses the unique ID to discover and track a VM. If your environment includes multiple VMs with the same ID, then discovery will assume they are the same VM. As a result, the counts for VMs will be incorrect.

## Supported Actions

For each discovered entity within the hypervisor supply chain, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type      | Can Be Automated                             | Recommendations Only              |
|------------------|----------------------------------------------|-----------------------------------|
| Virtual Machine  | Start, Move, Suspend, Resize Down, Resize Up | Terminate, Provision, Reconfigure |
| Physical Machine | Start, Suspend                               | Terminate, Provision              |
| Storage          |                                              | Provision                         |

## Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

- Virtual Machine
  - Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
    - Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
    - Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
    - Latency is the measurement of storage latency utilized by the entity.
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters

- Net
  - The utilization of data through the host's network adapters
- Swap
  - The utilization of the host's swap space
- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.
- Note:**
  - When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.
- Latency
  - The utilization of latency on the datastore.
- Datacenter
  - Note:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the hosts in the datacenter.
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use.
  - CPU
    - The utilization of the host's CPU reserved or in use.
  - IO
    - The utilization of the host's IO adapters.
  - Net
    - The utilization of data through the host's network adapters.
  - Swap
    - The utilization of the host's swap space.

## Creating A Service User Account

The service account Turbonomic uses to connect to a Hyper-V host must be an Active Directory domain account. The account must have full access to the cluster. To create such an account, execute the following command at a PowerShell prompt:

```
Grant-ClusterAccess <domain>\<service_account> -Full
```

Additionally, the service account must have specific local access rights on each host. The easiest way to grant Turbonomic the access it requires is to add the domain account to the `Local Administrators` group on each Hyper-V server.

Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account on every Hyper-V host.

### **Note:**

Turbonomic does not support Restricted User Accounts on Windows 2012 Hyper-V nodes.

To create a restricted service account on your Hyper-V hosts:

1. Add the service account to each of the following local groups:
  - WinRMRemoteWMIUsers\_\_ (or Remote Management Users)
  - Hyper-V Administrators
  - Performance Monitor Users

**Note:**

These groups are examples only. If your version of Windows Server does not include these groups, contact Technical Support for assistance.

2. Grant permissions to the service account.

In the WMI Management console, grant the following permissions to the service account:

- Enable Account
- Remote Enable
- Act as Operating System (For Windows 2016)

3. Configure the WinRM security descriptor to allow access by the service account:

- At a PowerShell prompt, execute `winrm configSDDL default`.
- In the "Permissions for Default" dialog box, grant the service account Read and Execute access.

## vCenter Server

VMware vCenter Server provides a centralized management platform for VMware hypervisors. To manage your VMware environment with Turbonomic, you specify a vCenter Server instance as a target. Turbonomic discovers the infrastructure that target manages, and links it into a supply chain to deliver application performance management.

### Prerequisites

- Target User Account

A user account that Turbonomic uses to connect to your vCenter Server and execute actions. For information about setting permissions for this account, see [Creating a User Account In vCenter Server](#).

### General Considerations

Before you configure a vCenter Server target, you should consider the following:

- **Linked vCenters:**

For linked vCenters, you must add each vCenter Server separately so Turbonomic can communicate with each vCenter Server through a separate API endpoint.

- **Restricting Turbonomic Access to Specific Clusters:**

When you add a vCenter Server target, Turbonomic discovers all of the connected entities that are visible, based on the target account that it uses to connect to the vCenter Server target. If you have clusters or other entities you want to exclude from discovery, you can use the vSphere management client to the role of the Turbonomic account to No access for the given entities.

- **Shared Datastores:**

If you add more than one vCenter Server target that manages the same datastore, you can enable or disable datastore browsing to discover wasted files on the shared datastore:

- Enable datastore browsing:

To properly enable browsing, you must turn on the **Enable Datastore Browsing** option in the target configuration for each vCenter Server target that manages the shared datastore.

- Disable datastore browsing:

If you don't want datastore browsing over shared datastores, you must turn *off* the **Enable Datastore Browsing** option in the target configuration for each vCenter Server target that manages the shared datastore.

If set **Enable Datastore Browsing** differently for separate targets that manage the same datastore, datastore browsing can give inconsistent results for active and wasted files.

- **VSAN Permissions:**

In order to enable VSAN support and discover groups based on storage profiles, you must ensure that the user role Turbonomic is assigned has the Profile-driven storage view permission enabled. This permission is *disabled* in the built-in readonly role.

- **Guest Metrics:**

**Enable Guest Metrics** is enabled by default for all vCenter Server targets added to Turbonomic.

**Note:**

If your vCenter Server target was added prior to 8.7.3, this option is enabled as part of the update.

This target option instructs that Turbonomic request from the VC instance that it start collecting and reporting enhanced memory metrics from each virtual machine, using the VMware Tools installed on each VM. Without this option, Turbonomic only has access to the "Active" memory for each virtual machine, which is a less accurate representation of the memory that the VM is really using.

For accurate metrics use vCenter Server 6.5 U3 or higher with VMware Tools version 10.3.2 or higher on guest VMs. If you run vCenter Server version 7.0 or greater, then it can automatically update your VMware Tools through the user interface. For other versions of vCenter Server, you must manually update your VMware Tools deployments.

For the permissions required for Guest Metrics, see [vCenter Permissions](#).

- **VMware Cloud Deployments:**

VMware Cloud (VMC) is a managed cloud offering that provides dedicated VMware vSphere-based Software Defined Data Centers (SDDC). VMC on AWS hosts SDDC on AWS facilities. For on-premise VMC, SDDC runs on specific hardware that meets VMware specifications.

VMware Cloud on AWS, Azure VMware Services, Google Cloud VMware Engine, IBM Cloud for VMware and other cloud hosted VMware solutions are treated like any other VMware integration. The same familiar Turbonomic experience can be expected when using the VMware vCenter Server target integration against any vCenter Server target whether it is cloud hosted or not, provided you pay close attention to satisfying all of the vCenter target prerequisites, especially permissions and roles. For example, VMware Cloud bare metal host instances may be modeled in Turbonomic as Hyperconverged templates, assigned a cost, and used to plan the migrating from existing on-premises host clusters to cloud hosted VMware solutions. Once in the cloud, you can use Turbonomic to plan migrations from one cloud hosted VMware solution to another. Provision and Suspend Host actions generated by Turbonomic may be more readily actionable on cloud hosted VMware solutions and automated using custom Action Scripts if required. If you have any requirements or improvements that relate specifically to cloud hosted VMware solutions, submit an enhancement request through [IBM Ideas](#) for future consideration.

For VMC environments, you can use the Turbonomic vCenter Server target to manage workloads, but you should be aware of the following concerns:

- DRS Settings

VMC SDDC requires DRS to be set to migration level 3. You cannot change this setting.

- Datacenter Names

VMC SDDC datacenter names are always SDDC Datacenter. This is true even for multiple VMC targets. You can use the Target Name filter to search and group VMs across the set of VMC SDDC datacenters and clusters that share this name.

- Discovery

- Turbonomic does not discover VMC service costs or subscription details.
- The user interface does not show cost savings or investments.
- Migration Plans
 

Plans do not discover VMC bare metal hosts. You can use Turbonomic HCI Host Templates to represent the bare metal instances. Then the plan can determine the number of VMC instances you need to support your existing on-premise workloads.

Migrate to Cloud plans do not support VMC.

## Adding vCenter Server Targets

To add vCenter Server targets, select the **Hypervisors > vCenter** option on the Target Configuration page and provide the following information:

- Address
 

The name or IP address of the vCenter Server.
- Username/Password
 

Credentials for the user account Turbonomic can use to connect to the vCenter Server. Include the domain if required (<domain>\<username>).
- Enable Datastore Browsing
 

Enabling datastore browsing enables Turbonomic to discover wasted storage.
- Enable Guest Metrics
 

Collecting advanced guest memory metrics can increase the accuracy of the VMEM data that Turbonomic uses for analysis of virtual machines. To enable guest metrics, ensure the following:

  - VMware Tools is installed and running on the target VMs.
  - The **Hypervisor VMEM for Resize** vCenter Server option is active for the VMs discovered by the target.
  - The user account has the `Performance.Modify Intervals` performance privilege.

For more information, see [Hypervisor VMEM for Resize](#) and [vCenter Performance Privileges](#).

### Note:

If your VMware hypervisors are running in a Nutanix environment, you must understand pinning a Nutanix Controller VM. For more information, see [Pinning Controller VMs in Generic Hypervisor Mode](#).

## vCenter Server Imported Settings

In addition to discovering entities managed by the hypervisor, Turbonomic also imports a wide range of vSphere settings, such as Host DRS rules, annotations, Resource Pools, and DRS HA settings (See [Other Information Imported From vCenter](#)).

**Note:** Turbonomic does not import Storage DRS rules at this time.

VMware vSphere 6.0 introduced the ability to move VMs between vCenters. If you enabled this feature in your VMware environment, you can configure Turbonomic to include cross vCenter Server vMotions in its recommendations.

To configure Turbonomic to support cross vCenter Server vMotion recommendations, you must create a Workload Placement Policy that merges the datacenters on the different vCenters, and then another policy to merge the given clusters. Also note that the merged clusters must use the same network names in the different datacenters. To create a Merge Policy:

1. In the Policy Management Tab, select **Placement Policy**.
2. For policy type, select **Merge**.
3. For MERGE, choose the merge type, and click **Select**.

To merge datacenters choose Datacenter, to merge Host clusters choose Cluster, and for storage choose StorageCluster.

4. Choose the specific datacenters or clusters to merge in this policy, then click **Select**.

5. Click **Save Policy**.

**Note:**

Since Turbonomic can only execute vMotions between clusters that use the same switch type (VSS or VDS), make sure any clusters you merge use the same switch type. Although Turbonomic will not initiate VSS → VDS vMotions, vSphere may do so. If this happens, Turbonomic displays a compliance violation notification.

## Monitored Resources

Turbonomic monitors the following resources for the hypervisor supply chain:

- Virtual Machine

- Energy

- Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

- Carbon Footprint

- Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

- Virtual Memory

- Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

- Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Storage (VStorage)

- Virtual Storage is the measurement storage utilized by the entity.

- Storage Access

- Storage Access is the measurement of IOPS utilized by the entity.

- Latency

- Latency is the measurement of storage latency utilized by the entity.

- Host

- Energy

- Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

- Carbon Footprint

- Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

- Memory (Mem)

- The utilization of the host's memory reserved or in use

- CPU

- The utilization of the host's CPU reserved or in use

- IO

- The utilization of the host's IO adapters

- Net

- The utilization of data through the host's network adapters

- Swap

- The utilization of the host's swap space
  - Balloon
    - The utilization of shared memory among VMs running on the host. ESX-only
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.
  - Note:**
    - When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.
  - Latency
    - The utilization of latency on the datastore.
- Datacenter
  - Note:** For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the hosts in the datacenter.
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use.
  - CPU
    - The utilization of the host's CPU reserved or in use.
  - IO
    - The utilization of the host's IO adapters.
  - Net
    - The utilization of data through the host's network adapters.
  - Swap
    - The utilization of the host's swap space.
  - Balloon
    - The utilization of shared of memory among VMs running on the host. ESX-only.
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only .
- Provider Virtual Datacenter
  - Memory (Mem)
    - The utilization of the Datacenter's memory reserved or in use.
  - CPU
    - The utilization of the Datacenter's CPU reserved or in use.
  - Storage

The utilization of the storage attached to the Provider vDC.

- Consumer Virtual Datacenter

- Memory (Mem)

The utilization of the Datacenter's memory reserved or in use.

- CPU

The utilization of the Datacenter's CPU reserved or in use.

- Storage

The utilization of the storage attached to the Consumer vDC.

## Actions

For each discovered entity within the hypervisor supply chain, Operations Manager can execute or recommend certain actions, as outlined below.

**Note:**

In order to execute cross-VC migrations as a non-admin user, you must have the following permissions enabled for the user account in both origination and destination vCenters:

- Virtual Machine: Edit Inventory, Create From Existing (Move, Register, Remove, Unregister sub-options), Create New
- Datacenter: Reconfigure Datacenter
- Network: Assign Network

| Entity Type      | Can Be Automated                                              | Recommendations Only              |
|------------------|---------------------------------------------------------------|-----------------------------------|
| Virtual Machine  | Start, Move, Move VM Storage, Suspend, Resize Down, Resize Up | Terminate, Provision, Reconfigure |
| Physical Machine | Start, Suspend                                                | Terminate, Provision              |
| Storage          |                                                               | Provision                         |

## Creating A Service User Account In vCenter

The service account you use must have specific permissions on the vCenter. The easiest way to grant Turbonomic the access it requires is to grant full administrator rights.

Some enterprises require that the service account does not grant full administrator rights. In that case, you can create a restricted service account that grants the following permissions to enable the required Turbonomic activities:

### vCenter Permissions

| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"> <li>• Read-only role for all entity types</li> </ul> <p>Assign either Global permissions or permissions for the given vCenter Server instance to the target user or user group.</p> <ul style="list-style-type: none"> <li>• Requirement to monitor VSAN and storage profiles</li> </ul> <p>In order to enable VSAN support and discover groups based on storage profiles, you must enable the Profile-driven storage</p> |



| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | view permission. This permission is <i>disabled</i> in the built-in readonly role.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Recommend Actions        | <ul style="list-style-type: none"> <li>• Read-only role for all entity types</li> </ul> Assign either Global permissions or permissions for the given vCenter Server instance to the target user or user group.                                                                                                                                                                                                                                                                                                       |
| Wasted Storage Reporting | <ul style="list-style-type: none"> <li>• Datastore &gt; Browse Datastore</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Execute VM Move          | <ul style="list-style-type: none"> <li>• Resources &gt; Assign VM to Resource Pool</li> <li>• Resources &gt; Migrate Powered Off VMs</li> <li>• Resources &gt; Migrate Powered On VMs</li> <li>• Resources &gt; Modify Resource Pool</li> <li>• Resources &gt; Query Vmotion</li> </ul>                                                                                                                                                                                                                               |
| Execute VM Storage Move  | <ul style="list-style-type: none"> <li>• Datastore &gt; Allocate Space</li> <li>• Datastore &gt; Browse Datastore</li> <li>• Resources &gt; Assign VM to Resource Pool</li> <li>• Resources &gt; Migrate</li> <li>• Resources &gt; Modify Resource Pool</li> <li>• Resources &gt; Move Resource Pool</li> <li>• Resources &gt; Query VMotion</li> <li>• Virtual Machine &gt; Change Configuration &gt; Change resource</li> <li>• Virtual Machine &gt; Change Configuration &gt; Change Swapfile placement</li> </ul> |
| Execute VM Resize        | <ul style="list-style-type: none"> <li>• Virtual Machine &gt; Change Configuration &gt; Change CPU count</li> <li>• Virtual Machine &gt; Change Configuration &gt; Change Memory</li> <li>• Virtual Machine &gt; Change Configuration &gt; Change resource</li> <li>• Virtual Machine &gt; Interaction &gt; Reset</li> <li>• Virtual Machine &gt; Interaction &gt; Power Off</li> <li>• Virtual Machine &gt; Interaction &gt; Power On</li> </ul>                                                                     |
| Discover Tags            | <ul style="list-style-type: none"> <li>• Global &gt; Global tag</li> </ul> You must also open ports 10443 and 7443 on the target server                                                                                                                                                                                                                                                                                                                                                                               |
| Guest Metrics            | <ul style="list-style-type: none"> <li>• Performance &gt; Modify Intervals</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### Other Information Imported from vCenter

In addition to discovering entities managed by the vSphere hypervisors and their resources, Turbonomic:

- Imports any vSphere Host DRS rules when DRS is enabled, and displays them on the **Policy > Workload Placement** view under **Imported Placement Policies**. Imported rules are enabled by default, but you can disable them in Turbonomic.

**Note:**

In vCenter environments, Turbonomic does not import DRS rules if DRS is disabled on the hypervisor. Further, if Turbonomic did import an enabled DRS rule and somebody subsequently disables that DRS rule, then Turbonomic will discover that the rule was disabled and will remove the imported placement policy.

- Imports any custom annotations and displays related groupings in the **Inventory > Groups** tree view, under **VC Annotations**. The service account must enable the **Global > Global tag** privilege, and the target server must open ports 10443 and 7443.
- For vCenter Server versions 5.5 and later, discovers Virtual Machine Storage Profiles and displays them as groups anywhere that you can set scope. The groups appear under **VC Storage Profiles**. You can use these discovered storage profiles the same as any other groups — For example, to scope dashboards, or to set the scope for specific action policies.
- Discovers resource pools and displays them as folders in the Inventory tree and as components in the Supply Chain Navigator. If you have the Cloud Control Module license, Turbonomic manages resource pools as Virtual Datacenters (VDCs) and can recommend resize actions. Root resource pools appear as Provider VDCs in the supply chain, whereas child resource pools appear as Consumer VDCs.
- Imports vSphere HA cluster settings and translates them into CPU and memory utilization constraints. These are displayed as cluster-level overrides under **Folders** on the **Policy > Analysis > Host** view.

## Orchestrator Targets

---

Turbonomic supports Action Script and ServiceNow orchestrator targets.

With orchestrator targets you can integrate Turbonomic actions with the orchestrator's application management process. For example, you can pass Turbonomic to a Change Request system for approval, and the system can pass the action back to Turbonomic for execution.

**Note:**

Turbonomic uses remote servers to execute ActionScript orchestration. Managing the processes remotely means that you do not install custom code on the Turbonomic server. This eliminates associated security risks at that point. However, you are responsible to maintain the security of your Action Script Server, and to ensure the integrity of your custom code.

## ActionScript Server

### Resource Requirements for the Server

The remote server can be a VM or a container. The capacity you configure for the server depends entirely on the processes you intend to run on it. Turbonomic does not impose any special resource requirements on the server.

### Configuring Command Execution

To support execution of your scripts, you must install any software that is necessary to run the scripts. This includes libraries, language processors, or other processes that your scripts will invoke.

Turbonomic invokes the scripts as commands on the server. The server must run an SSH service that you have configured to support command execution and SFTP operations. At this time, IBM has tested action scripts with the OpenSSH sshd daemon.

The standard port for SSH is 22. You can configure a different port, and provide that for admins who configure the server as an ActionScript target.

Note that an action script can invoke any process you have deployed on the remote server. You do not have to run scripts; however, you must be able to invoke the processes from the command line. The script manifest gives Turbonomic the details it needs to build the command line invocation of each script.

## Configuring the ActionScript User Account

To execute the scripts on your server, Turbonomic logs on via a user account that is authorized to execute the scripts from the command line. You provide the user credentials when you configure the ActionScript target. To support this interaction, the user account must meet the following requirements:

- Public Key

The user must have a public key in the `.ssh/authorized_keys` file. When you configure the ActionScript target, you provide this as the Private Token for the target.

### Note:

For the ActionScript target to validate, the permissions on the `authorized_keys` file must be set to 600.

- Security for the `.ssh` Directory

The ActionScript User should be the only user with authorized access. You should set file permissions to 700.

- Supported Shells

The ActionScript User shell can be either the Bourne shell (usually at `/bin/sh`) or the Bourne-Again shell (usually at `/bin/bash`). Turbonomic passes parameters as it invokes your scripts. At this time it only supports script execution through these shells.

## Handling ActionScript Timeouts

Turbonomic limits script execution to 30 minutes. If a script exceeds this limit, Turbonomic sends a SIGTERM to terminate the execution of the process.

Note that Turbonomic does not make any other attempt to terminate a process. For example you could implement the script so it traps the SIGTERM and continues to run. The process should terminate at the soonest safe opportunity. However, if the process does not terminate, then you must implement some way to terminate it outside of Turbonomic. Note that a runaway process continues to use its execution thread. This can block other processes (action scripts or primary processes) if there are no more threads in the pool.

## Obtaining Private Key String

In order to pass the private key as a string, newlines must be inserted appropriately with the `\n` character, as the JSON format will not preserve these required newlines. One method to insert these lines on \*nix systems is the `awk` command. This command should follow the following format: `awk 'ORS="\n" ' path/to/keyfile > path/to/modified/keyfile`. For example, `awk 'ORS="\n" ' ~/.ssh/id_rsa > /tmp/privkey.txt`. You can then `cat` or `vi/vim` the file to copy the text and insert it into the API call.

Example private key file, before modification:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEJKQIBAAKCAgEAxeKUK6WLFMf8seI2KztL4DoKKw0d6AG/39SH6or5SiNbAGkM
Umv8ABA0aCQqMF01j9V3kzGn71DIWWTxLOMBH06I01dvd9XYPVRJotGjpxus+xmV
GbeXxJsdHfXrbtMeLnpkwGC7gIu6W2o5Ez6YuLdkf+CeFZhGM0tVLiB2Nwtzn33D
KBh96tIom9EqKseovtQAj5VYDwPeuTawX1RGMt11tsNGyezEc4Z2sppT4VfVhsUV
L+voDZzrtMfFiHUigjPH4pK9c/iPj0um7YzmyMyARftyCsUg7iUZH4N1NLPikXVy
xKiro6Quu7SZ4kcvr3ZF3970CON0hHzC3mFsvZvRFgYKzgc8/BcZ8ZzEUFfqBqY4
dejCU+jo86dYFL2Q8RfF2zsJenG0WbVRLtLGkc7RbMdBGjtt5roV5VAp+TyR9Hdd
Dc5FYvt4fAAMJtcmTIzdEAPsbVzDhkSajxRh0XvezJn+2Ddr34/9ZvFE7nJNgP/8
KOGJ5H1dgG5mc7H+Ly3x5dXzPhcVNSUdNGgKnevzFgMJaGByAg5AZc5H0miIwFKs
KSE/m8dSR0HrHmenGgmk1IMUnvCC1mKkbUEJ2V6R17x8zmM4DYNRSTwcSL
SZ7U/yDuwpHwzeozNeoekGKeC23+jgyzr9/mzgmPF08q5F+wVg/7n1DRSXsCAwEA
```

```
AQKCAgBPHMPbt25AM6gCEfcoF3/BOFMCXS8cpz5npRd1nIQHsNTb+CN1/Df2aoQZ
MoQva8RTKbuxfE0weLJwktuc33B+bK51F+q31SXrdvwtfPSZhYFFqXejnX0upXjh
1oS2f07vE/2dr86mmlY7f98ro3X50KzsrQdmiMdfHWaj11TLdJdgJy5TpF8K91RG
3zuleMcs66MBxGo9LoXntDwwBog9e8Rk+fMeHwt46EQiKXLPfNPdmoNa8kyGzLqr
PsbpRCjPrRX3rkTRzYjPeCZXiGJn2uJZmCpCxcgbhTrGZmLVofC+WcCefeg2M0cHh
R1XSUjVtR7KtyvzvZcQah6p8r6SCASQFeVbb13FJVkBd0J8Hzk5e4N5Qk3bv9V4W
F7CC8dRV66kAkf68ytLGXkQASgo5RPBDYSp8E2JDLF4m+9w6aouPlg//QxG5+XA
wqGJseTipiHbiAlcIGsCpc5QCqab2SNddaDtIIPnp2Zmo5E10jutGeUAE3k+cnJ1
JIFCkRL0vsPk0xODSXbV2PMAUjMT65ue7uBkDqCtgHiSBXTk6DiWBjduDnoZ1AmC
I8M4o42mSPs2uoKcngJ2fboY5iSsXLQ8VT5Tix8fk7VzBFm0vchBUSFlyx6o3/01
0SV3Qrqz7tb1Xx+axQapIUEVbLV140K0jepvxF3HVELXRQD2eQKCAQEA8EzXro5j
N/VgDQJ9pGphwUyQ9mNBC1c1jviAHHskh8gwa4BxhBLzZ+aUuoIx0/mAYK0rihN+
QJL184c9tA62Au6Tq+CT60ad4SwxQ5m9mQZjq888T3dRssmY+f+5kYBzmaPHv18I
Sohh2tHqNh68T1KmlbCABahc0ppqJTjp00i0kR7XUk5gEeDNNb65oD0tuNytiVd4
cNeq/c08rqk2rTupLLjhVKzP91gVXL/SBw/HBjpmYJ3DmrcPXH84JemshJay/Onw
eY3Y+hsghw37zRxeuUtn5to0wVYayAmEhyXVing7Evv5wsyUaJQE25ZZWLjjiBHP
ZpLsSRc++uugnwKCAQEA0tBSQRDd0H+06Ydu9Tno1IUEilTQ8+VGx3esFlKtiDTW
cSEZsfZKbf/GQgHKbXlVXjCLhbfPbCvCcojbmGSmtWRWxlySYPsw4tBiYJsk6Fo8
J2hEYw3HQzyj23RjXIghCUqY5wNwj1brU4CRxHtMyV1eYJWgmTaBfDm8VoBghu9g
fAYPkJs4WZuKI7Sbg2vcJyLLP4XETPlfACMptLrPs8RCyUjJAJYxpsdaSjLc7r9
9gBusQ01IgpClrx+/86NYhdKDWmnd0lyjXNvMVwPipoSvblXTZK7xRBgcNTsd2eh
h1HLuhLXZ76SplvtNH6MQP46Wob1ounEx70NnsddpQKCAQEA11VD339DQktLo+7/
0CHPFZHMjueSwaHEtMQzyLORrsbBYXGMTqh0kM0aqM1V1mNpvVFYt2z5pHEIaibL
w4mHk0/pqVnDtGwjK9R1aEQeeg3SKysd5j0tzCkX0mIIC5G2X5KHUtoQWsePHX
XPRnND1c2c3fMqysNndiklatDgIx5Grk1ZxwKgtH+2ZnIpCHmw/IQxy8fWvae8n
oYzurW9Ugv5YhaUqT0jDeP9PP1s0QvriHN/13eQnM2szicPDC6XRdK7cKxRrFwJ3P
kobHEHh8ilUyblF0sGRW7EZk04MvNhmnnxLYXoG0Y01FBNz1+45jEm0e0g7Fypz
021NswKCAQAHx/6xTXvs4PByeDhFPuz0IzG8Y2MPDZBuPmGQtV6IX4LmzBT140xg
YQd9WjwRnfi1Yk6yKs/0o/fiAZ8LLPjPVEc4Pxxkz03TJQjuaCsBniJaqAvd2ARYAK
wEtAB2y/dprZj/JS3JGuWsp24hd/UpnD+PLJ0VsVfqiK4EquwrC1a1J+ivQ1n0UW
MwYUvyDPRRwLqQu0Dpb1YKGAyrhmU/Sw7c2wPp+sbZw/Rq8Euae8+Br1oJB1YjAP
ITzx+wUqpqIb88Yc9qBiy8g5pmAdT0uTuGi13eN2Dot1u6QDpiphQ0/f3WbKoxLGS
t6EMG0qJfH5Ky9d1jfmD70j/IgaBLNstAoIBAQCkqJNuH7Fsta/KQ5fJzFrD/Mil
Q6yF8D92UEab1Et22zr7Ry5WAv4DRcYwV1HrBjQB0w4Gc5WzCEnE1HvhkX519w
hccp1P+tyRf+h6ppYqd1/Yaw2vEMTR1gvvtV8+2aMYTNRQJuTjJIsk1vWvX42IhQ
xqKM9xmqKGEsw+6e7UheaXY8afkjjf20/euGgzTuEPoy+IA8vKzSanLlWm4xDYjMV
i9PFbuSyRV6tJqQZxZ58dS2ykL5YpTLM+ZFYzmaAqSjWULpsKAXxE2fnmniXOYVF
E4VxEPsgNYH1k2eqtGRQwYR0ML/fihHbJIIQnMr6o7/tK1jx+3PCZeRavR4/
-----END RSA PRIVATE KEY-----
```

The same private key, after running the above command:

```
-----BEGIN RSA PRIVATE KEY-----\n
MIIJKQIBAAKCAgEAxeKuk6WLFmf8seI2KztL4DoKkWoD6AG/39SH6or5SiNbAGkM\n
Umv8ABA0aCqMF01j9V3kzGn71DIWWTxLOMBH06I01dvd9XYpVRJotGjpxus+xmV\n
GbeXxJsdHfXrhtMeLnpkwGC7gIu6W2o5EzYulDKf+CeFZhGM0tVLiB2Nwtzn33D\n
KBh96tIom9EqKseovtQAj5VYDwPeuTaWX1RGMt11tsNGyezEc4Z2sppT4VfVhsUV\n
L+voDZzrtmFfIHUIgJPH4pK9c/iPjOum7YzmyMyARfityCsUg7iUZH4N1NLpkXV\n
xKiro6Quu7SZ4kcvr3ZF3970C0N0hHzC3mFsvZvRFgYKzgc8/BcZ8ZzEUFfqBqY4\n
dejCU+jo86dYFL2Q8Rf2zsJenG0WbVRLtLgkC7RbMdBGjtt5roV5VAp+TyR9Hdd\n
Dc5FYvt4fAAMJtcmTIzdEAPsbVzDhkSajxRh0XvezJn+2Ddr34/9ZvFE7nJNgP/8\n
K0GJ5Ht4fG5mc7H+Ly3x5dXzPhcVNSUdNgGKnEvzFgMJaGByAg5AZc5H0miIwFKs\n
KSE/m8dSR0HrHmenGGmk1IMUnvCC1mKkbUEJ2V6R17x8zmM4DYNRSTwCSL\n
SZ7U/yDuwpHwzeozNeoeKGeK23+jgyzr9/mzgmPF08q5F+wwVg/7n1DRSxScaWEA\n
AQKCAgBPHMPbt25AM6gCEfcoF3/BOFMCXS8cpz5npRd1nIQHsNTb+CN1/Df2aoQZ\n
MoQva8RTKbuxfE0weLJwktuc33B+bK51F+q31SXrdvwtfPSZhYFFqXejnX0upXjh\n
1oS2f07vE/2dr86mmlY7f98ro3X50KzsrQdmiMdfHWaj11TLdJdgJy5TpF8K91RG\n
3zuleMcs66MBxGo9LoXntDwwBog9e8Rk+fMeHwt46EQiKXLPfNPdmoNa8kyGzLqr\n
PsbpRCjPrRX3rkTRzYjPeCZXiGJn2uJZmCpCxcgbhTrGZmLVofC+WcCefeg2M0cHh\n
R1XSUjVtR7KtyvzvZcQah6p8r6SCASQFeVbb13FJVkBd0J8Hzk5e4N5Qk3bv9V4W\n
F7CC8dRV66kAkf68ytLGXkQASgo5RPBDYSp8E2JDLF4m+9w6aouPlg//QxG5+XA\n
wqGJseTipiHbiAlcIGsCpc5QCqab2SNddaDtIIPnp2Zmo5E10jutGeUAE3k+cnJ1\n
JIFCkRL0vsPk0xODSXbV2PMAUjMT65ue7uBkDqCtgHiSBXTk6DiWBjduDnoZ1AmC\n
I8M4o42mSPs2uoKcngJ2fboY5iSsXLQ8VT5Tix8fk7VzBFm0vchBUSFlyx6o3/01\n
0SV3Qrqz7tb1Xx+axQapIUEVbLV140K0jepvxF3HVELXRQD2eQKCAQEA8EzXro5j\n
N/VgDQJ9pGphwUyQ9mNBC1c1jviAHHskh8gwa4BxhBLzZ+aUuoIx0/mAYK0rihN+\n
QJL184c9tA62Au6Tq+CT60ad4SwxQ5m9mQZjq888T3dRssmY+f+5kYBzmaPHv18I\n
Sohh2tHqNh68T1KmlbCABahc0ppqJTjp00i0kR7XUk5gEeDNNb65oD0tuNytiVd4\n
cNeq/c08rqk2rTupLLjhVKzP91gVXL/SBw/HBjpmYJ3DmrcPXH84JemshJay/Onw\n
eY3Y+hsghw37zRxeuUtn5to0wVYayAmEhyXVing7Evv5wsyUaJQE25ZZWLjjiBHP\n
ZpLsSRc++uugnwKCAQEA0tBSQRDd0H+06Ydu9Tno1IUEilTQ8+VGx3esFlKtiDTW\n
cSEZsfZKbf/GQgHKbXlVXjCLhbfPbCvCcojbmGSmtWRWxlySYPsw4tBiYJsk6Fo8\n
J2hEYw3HQzyj23RjXIghCUqY5wNwj1brU4CRxHtMyV1eYJWgmTaBfDm8VoBghu9g\n
fAYPkJs4WZuKI7Sbg2vcJyLLP4XETPlfACMptLrPs8RCyUjJAJYxpsdaSjLc7r9\n
9gBusQ01IgpClrx+/86NYhdKDWmnd0lyjXNvMVwPipoSvblXTZK7xRBgcNTsd2eh\n
h1HLuhLXZ76SplvtNH6MQP46Wob1ounEx70NnsddpQKCAQEA11VD339DQktLo+7/\n
0CHPFZHMjueSwaHEtMQzyLORrsbBYXGMTqh0kM0aqM1V1mNpvVFYt2z5pHEIaibL\n
w4mHk0/pqVnDtGwjK9R1aEQeeg3SKysd5j0tzCkX0mIIC5G2X5KHUtoQWsePHX\n
```

```

XPRnND1c2c3fMqysNnDiklatDgIx5Grk1ZXwnKGth+2ZNipCHmw/IQxy8fwvae8n\n
oYzurW9Ugv5YhaUqT0jeDP9PP1s0QvriHN/13eQnM2szicPDC6XRdK7cKxRrFWJ3P\n
kobHfHh8iLuYb1Fc0sGRW7EZk04MvNhmxx1YXoG0Y01FBNz1+45jEm0e0g7Fypz\n
021NswKCAQAHx/6xTXvs4PByeDhFpuz0IzG8Y2MPDZBuPmGQtV6IX4LmzBT140xg\n
YQd9WjmRnF1YK6yKs/0o/fiAZ8LLPjPVEC4Ppkz03TJQjuaCsBniJaqAvd2ARyAK\n
wEtAB2y/dprZj/JS3JGuWsP24hd/UpnD+PLJ0VsVfqiK4EquWrC1a1J+ivQ1n0UW\n
MWYUvyDPRRwLqQu0Dpb1YKgAyrrhmU/Sw7c2wPp+sbZW/Rq8Euae8+Br1oJB1YjAP\n
ITzx+WUqpgIb88Yc9qBiY8g5pmAdT0uTuGIi3eN2Dot1u6QDpihQQ/f3WbK0XLGs\n
t6EMG0qJfH5Ky9d1jfmD70j/IgaBLNstAoIBAQCkqJNuH7Fsta/KQ5fJzFrD/Mil\n
Q6yfb9D92UEabB1Et22Zr7Ry5WAv4DRcYwV1HrBJQB0w4Gc5WZcEnE1HvhkX519w\n
hccp1P+tyRf+h6ppYqd1/Yaw2vEMTR1gvvtV8+2aMYTNRQJuTjJIsk1vWvX42IhQ\n
xqKM9xqmKGEsw+6e7UheaXY8afkjf20/euGgzTuEPoy+IA8vKrSAnLLWm4xDYjMV\n
i9PFbuSyRV6tJqQZxZ58dS2ykL5YpTLM+ZFyZmhAqSJWULpsKAxxE2fnmniX0YVF\n
E4VxEPsgNYH1k2eqtGRQwYR0ML/fihHbJIIQnMr6o7/tK1jx+3PCZeRaVR4/\n
-----END RSA PRIVATE KEY-----\n

```

## Adding ActionScript Targets to Turbonomic

### Note:

At this time, ActionScript targets must be added using the API. The ActionScript target appears in the UI as a selection, but should not be used. For more information about the Turbonomic API and how to use it to add targets, see [API Reference](#).

The TargetApiInputDTO for this target has the following parameters:

#### nameOrAddress

IP or FQDN of the script execution server.

#### port

Port used to connect to the script execution server.

#### userid

Username used to connect to the script execution server.

#### privateKeyString

Path to the SSH private token corresponding to the user used to connect to the script execution server.

#### manifestPath

Path to the ActionScript manifest file on the script execution server.

Use the POST `https://10.10.10.10/api/v3/targets` request to add this target. Send the constructed TargetApiInputDTO as the body of the request.

### Example Input:

```

{
 "category": "Orchestrator",
 "inputFields": [
 {
 "name": "nameOrAddress",
 "value": "10.10.10.10"
 },
 {
 "name": "port",
 "value": "22"
 },
 {
 "name": "userid",
 "value": "shepp"
 },
 {
 "name": "privateKeyString",
 "value": "-----BEGIN RSA PRIVATE KEY-----\nMIIJKQIBAACKAgEAxeKUK6WLFMf8seI2KztL4DoKKw0d6AG/
39SH6or5SiNbAGkM\nUmv8ABA0aCQqMF01j9V3kzGn71DIWwTxLOMBH06I01dvd9XYPVRJotGjpxus+xmV\nnGbeXxJsdHfXr
btMeLnpkwGC7gIu6W2o5Ez6YuLDKf+CeFZhgM0tVLiB2Nwtzn33D\nnKbH96tIom9EqKseovtQAj5VYDwPewTaWx1RGMT11ts
NGyezEc4Z2sppT4VFVhsUV\nnL+voDZzrtMfFiHUIgjPH4pK9c/
ipj0um7YzmyMyARftyCsUg7iUZH4N1NLPikXVY\nnxKiro6Quu7SZ4kcvr3ZF3970CON0hHzC3mFsvZvRFgYKzgc8/
BcZ8ZzEUFfqBgY4\nndeJCU+jo86dYFL2Q8RfF2zsJenG0WbVRLtLGkc7RbMdBGjtt5roV5VAp+TyR9Hdd\nnDc5Fvvt4fAAMJ
tcmTIzdEAPSBvZDhkSajxRhOXvezJn+2Ddr34/9ZvFE7nJNgP/
8\nnKOGJ5H1dgg5mc7H+Ly3x5dXzPhcVNSUDNGgKnEvzFgMJAgByAg5AZc5H0miIwFKs\nnKSE/
m8dSRoHzHmenGGm5TtA41u1IMUnvCC1mKkbUEJ2V6R17x8zmM4DYNRSTwcSL\nnSZ7U/

```

```

yDuwpHwzeozNeoekGKeC23+jgyzr9/mzgmPF08q5F+wVg/7n1DRSXsCAwEA\nAQKCAgBPHMPbt25AM6gCEfcoF3/
BOFMCXS8cpz5npRdlnIQHsNTb+CN1/
Df2aoQZ\nMoQva8RTKbuxfE0weLJwktuc33B+bK51F+q31SXrdvwtfPSZhYYFqXejnX0upXjh\nloS2f07vE/
2dr86mLly7f98ro3X50KzsrQdmiMdfHwajl1TLdJdgJy5TpF8K91RG\n3zuleMcs66MBxGo9LoXntDwwBog9e8Rk+fMeHwt4
6EQiKXLPfNPdmoNa8kyGzLqr\nPsbpRCjPrRX3rkTRzYjPeCZXiGJn2uJzMcPcxgbhTrGZmLVofC+Wccefeg2M0cHh\nnR1XS
UjVtR7KtyvvzZcQah6p8r6SCASQFeVbb13FJVkBD0J8Hzk5e4N5Qk3bv9V4W\nf7CC8dRV66kCAkF68ytLGxkQAsgo5RPBDY
Sp8E2JDLF4m+9w6aouPlg//
QxG5+XA\nnwqGJseTipiHbiALcIGsCpc5QCqab2SNddaDtIIPnp2Zmo5E10jutGeUAE3k+cnJ1\nnJIFCkRL0vsPk0x0DSXbV2
PMAUjMT65ue7uBkDqCtgHiSBXTk6DiWBjduDnoZ1Amc\nI8M4o42mSPs2uoKcngJ2fboY5iSsXLQ8VT5Tix8fk7VzBFm0vch
BUSFLy6o3/01\n0SV3Qrqz7tb1Xx+axQapIUEVbLV140k0jepvxF3HVELXRQd2eQKCAQEAE8EzXro5j\nn/
VgDQJ9pGphwUyQ9mNBC1c1jviAHHskh8gwa4BxhBLzZ+aUuoIx0/mAYK0rihN+
\nQJLI84c9tA62Au6Tq+CT60ad4SwxQ5m9mQZjq888T3dRsmY+f+5kYBzmaPHv18I\nnSohh2tHqNh68T1KmlbCABahc0ppq
JTjp00i0kr7XUK5gEeDNNb65oD0tuNytiVd4\ncNeq/c08rqk2rTupLLjhVKzP9lgVXL/SBw/
HBjpMyJ3DMrcPXH84JemshJay/
Onw\nneY3Y+hsghw37zRxeuUtn5to0wVYayAmEhyXVing7EVV5wsyUaJQE25ZZWLjjiBHP\nnZpLsRrc+
+uugngKCAQEAtBSQrDdOH+06Ydu9Tno1IueiltQ8+VGx3esFlKtiDTW\ncSEZSfZKbf/
GQgHKbxLVXjCllhbfpBcCvCcojbmGSmtWRWxlySYPSW4tBiYJsk6Fo8\nnJ2hEYW3HQzyj23rjXIghCUqY5WnWj1brU4CRxHtMy
V1eYJWgmTaBfDm8VoBghu9g\nfAYpKJs4WZUKUI7Sbg2vcJyLLP4XETPlfACMptLrPs8RCyUjJAJYxpSdaSjLC7r9\nn9gBus
Q0IIGpClrx+/
86NYhdKDWMnd0lyjXnVMVPIpoSVb1XTZK7xRBgcNTsd2eh\nnh1HLuhLXZ76Sp1vtNH6MQP46WoblounEx70NnsddpQKCAQE
A11VD339DQktLo+7/\ni0CHPFZHMjueSwaHEtMQzyLORrsbBYXGMTqh0kM0aqM1V1mNpvVFYt2z5pHEIaibL\nnw4mHk0/
pqVnDtgwjkA9R1aEQeqq3SKysd5j0tzCXkQ0mIIC5G2X5KHUt0QWSepHX\nnXPRnND1c2c3fMqysNnDiklatDgIx5Grik1ZXwn
KGth+2ZnipChmw/IQxy8fwvae8n\nnoYZurW9Ugv5YhaUqTOjeDP9PP1s0QvrHN/
13eQnM2szicPDC6XRdK7cKxRrFwJ3P\nnkobHfHh8iLuYb1Fc0sGRW7EZk04MvNhmnx1YXoGOY01FBNz1+45jEm0e0g7Fypz
\n021NswKCAQAHx/6xTXvs4PByeDhFPuz0IzG8Y2MPDZBuPmGQtV6IX4LmzBT140xg\nnYQd9WjmRnf1YK6yKs/0o/
fiAZ8LLPjPVEc4Pxxz03TJQjuaCsBniJaqAvd2ARyAK\nnwEtAB2y/dprZj/JS3JGuWsP24hd/
UpnD+P130VsVfqiK4EquWrC1a1J+ivQln0UW\nmWYUvyDPRRwLQQu0Dpb1YKgAyrhmU/Sw7c2wPp+sbZW/
Rq8Euae8+Br1oJB1YjAP\nnITzx+WUqpgIb88Yc9qBiy8g5pmAdT0uTuGi3eN2Dot1u6QDpihQQ/
f3WbK0XLGs\nnt6EMG0qJfH5Ky9d1jfmD70j/IgaBLNStAoIBAQCkqJNuH7FSta/KQ5fJzFrD/
Mil\nnQ6yfb9D92UEabB1Et22Zr7Ry5WAv4DRcYwV1HrBJQB0w4Gc5WZcEnE1HvhkX519w\nnhccp1P+tyRf+h6ppYqd1/
YaW2vEMTR1gvtv8+2aMYTNRQJUjJIsk1vWvX42IhQ\nnxqKM9xqmKGEsw+6e7UheaXY8afkjf20/
euGgzTuEPoy+IA8vKrsAnLLWm4xDYjMv\nni9PFbuSyRV6tJqQZx5Z8dS2ykL5YpTLM+ZFyZmhAqS3WULpsKAXxE2fnmniXOY
VF\nnE4VxEPsgNYH1k2eqtGRQWYR0ML/fihHbJ1lQnMr6o7/tk1jx+3PCZeRavR4/\n-----END RSA PRIVATE KEY-----
\n"
 },
 {
 "name": "manifestPath",
 "value": "/home/shepp/action-script/manifest.json"
 }
],
"type": "ActionScript"
}

```

The API will return the TargetApiDTO for the ActionScript target upon success, or an error message upon failure.

### Example Response:

```

{
 "uuid": "73356329749216",
 "displayName": "10.10.10.10-/home/shepp/action-script/manifest.json",
 "category": "Orchestrator",
 "inputFields": [
 {
 "displayName": "Name or Address",
 "name": "nameOrAddress",
 "value": "10.10.10.10",
 "isMandatory": true,
 "isSecret": false,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "IP or FQDNS for the Script Execution Server",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Port",
 "name": "port",
 "value": "22",
 "defaultValue": "22",
 "isMandatory": false,
 "isSecret": false,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "Port to use for the Script Execution Server",
 "verificationRegex": ".*"
 }
],
 "displayName": "User ID",
}

```

```

 "name": "userid",
 "value": "shepp",
 "isMandatory": true,
 "isSecret": false,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "Userid to use to execute command on the Script Execution Server",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Private Token",
 "name": "privateKeyString",
 "isMandatory": true,
 "isSecret": true,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "SSH Private Token corresponding to the Userid",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Script Path",
 "name": "manifestPath",
 "value": "/home/shepp/action-script/manifest.json",
 "isMandatory": true,
 "isSecret": false,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "File Path to the ActionScript manifest file on the Execution Server",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Public Host Key",
 "name": "hostKey",
 "isMandatory": false,
 "isSecret": false,
 "isTargetDisplayName": false,
 "valueType": "STRING",
 "description": "Public key presented by the SSH server for host authentication; if not
provided, the presented key will be accepted and integrated into the target definition for
future operations",
 "verificationRegex": ".*"
 }
],
"lastValidated": "2020-07-28T20:46:22Z",
"status": "Validated",
"type": "ActionScript",
"readonly": false
}

Response headers

cache-control: no-cache, no-store, max-age=

```

Once the success response is received, the target is validated and ready for use.

## Flexera One

Turbonomic integrates its management of workloads with Flexera One License Management. To do this, Turbonomic discovers the licenses and entitlements that you have configured in your Flexera environment. Turbonomic then creates groups and policies to represent these licenses and entitlements.

- For each license, Turbonomic creates a dynamic group.

These groups contain the entities that are affected by a Flexera license. For example, Turbonomic might create one group of all the VMs affected by a license, and another group of all the host machines affected by that license.

These groups are dynamic – If you allocate or remove entities (workloads, hosts, etc.) under a given license, Turbonomic adds or removes those entities in the given group. Even if you have a license with no entities allocated to it, Turbonomic creates an empty group for that license.

The name of each group begins with Flexera. The group name then includes the name of the license, plus the entity type. For example, you might see the groups, Flexera: Microsoft SQL Server 2016 Standard VMs and Flexera: Microsoft SQL Server 2016 Standard Hosts.

- For supported licenses, Turbonomic creates Placement and Automation policies.

A Flexera license can include placement constraints, or resource utilization constraints. Turbonomic can create policies to comply with those constraints:

– Placement Policies

When calculating workload placement, Turbonomic uses placement policies to comply with constraints such as keeping workloads on licensed hosts, or clearing unlicensed workloads off of licensed hosts when necessary.

For more information, see [Placement Policies](#).

– Automation Policies

Among other settings, these policies can specify operational, utilization, and scaling constraints to match the constraints in your Flexera license policies.

For more information, see [Automation Policies](#).

The name for each policy begins with the name Flexera. The policy name then includes the name of the license and constraint. For example, you might see a policy named Flexera: Microsoft SQL Server 2016 Standard restrict to hosts.

Turbonomic then applies these policies to the groups that correspond to the entity type and license name.

## Turbonomic Policies for Flexera Licenses

Turbonomic generates policies to comply with the following types of Flexera licenses, where **Usage Rights** lists the configurations Turbonomic considers when creating policies:

| License:                          | Usage Rights:                                                                                                                                                                                 | Notes:                                                                         |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Microsoft Server Processor        | Datacenter Edition<br><pre>"coverInstallsOnVirtualMachines": true, "limitNumberOfVirtualInstalls": false, "limitVirtualInstallsIncludesHost": true, "numberOfAllowedVirtualInstalls": 1</pre> | Create Placement Policies that limit on-prem VM moves to the licensed host.    |
|                                   | Standard Edition<br><pre>"coverInstallsOnVirtualMachines": true, "limitNumberOfVirtualInstalls": true, "limitVirtualInstallsIncludesHost": false, "numberOfAllowedVirtualInstalls": 1</pre>   | Create Placement Policies that disable VM move actions.                        |
| Microsoft Server/ Management Core | Datacenter Edition<br><pre>"coverInstallsOnVirtualMachines": true, "limitNumberOfVirtualInstalls": false, "limitVirtualInstallsIncludesHost": false</pre>                                     | Create Placement Policies that limit on-premise VM moves to the licensed host. |
|                                   | Standard Edition<br><pre>"coverInstallsOnVirtualMachines": true, "limitNumberOfVirtualInstalls": true, "limitVirtualInstallsIncludesHost": false, "minimumNumberOfProcessors": 2</pre>        | Create Placement Policies that disable VM move actions.                        |



| License:              | Usage Rights:                                                                                                                                                         | Notes:                                                                                                                                                                                                                                                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Microsoft Server Core | <pre>"coverInstallsOnVirtualMachines": false, "limitNumberOfVirtualInstalls": false, "limitVirtualInstallsIncludesHost": true, "minimumNumberOfThreadsPerVM": 4</pre> | <p>Turbonomic analysis assumes licensing is based on the host.</p> <p>Create Placement Policies that limit on-prem VM moves to the licensed host. Respect minimum vCPU thresholds.</p> <p>If there is a value for <code>minimumNumberOfThreadsPerVM</code>, create Automation Policies to enforce that minimum vCPU threshold for VMs.</p> |

## Flexera Permissions

The account that Turbonomic uses to connect with a Flexera target must be a Read-Only account with the following roles:

- View IT assets
- call APIs

## Adding Flexera Targets

You can configure a target to a Flexera SaaS account. To add a Flexera target, select the **Orchestration > Flexera** option on the Target Configuration page and provide the following information:

- Flexera Region
 

The region for your Flexera account. Choose one of:

  - NAM
  - EU
- Organization ID
 

The unique identifier that defines your organization within Flexera One.
- Refresh Token
 

The credential that Turbonomic can use to generate Access Tokens as needed.
- Proxy Host
 

The address of the proxy used for this target. Only fill out proxy information if you connect to the Flexera instance via a proxy.
- Proxy Port
 

The port to use with the proxy specified above. By default, this is 8080.
- Proxy Username
 

The username to use with the proxy specified above.
- Proxy Password
 

The password to use with the proxy specified above.
- Secure Proxy Connection
 

When checked, Turbonomic will connect to the proxy via HTTPS.

## Actions

Turbonomic does not generate unique actions for entities it discovers through a Flexera target. Instead, it uses the license specifications it discovers in Flexera to prevent actions that do not comply with license restrictions. It can also modify or generate actions to ensure license compliance.

When Turbonomic generates an action, the action details include a list of policies that constrain the action recommendation. In this list, policy names that begin with `Flexera:` indicate policies that implement the license constraints Turbonomic has discovered for Flexera. You can click to navigate to the policy definition. Note that you cannot see the policy definition, but you cannot edit it.

Move Virtual Machine Tomcat from [Host] to [Host]  
CPU Congestion

VIRTUAL MACHINE  
Tomcat

CURRENT HOST - IMPACT FROM ALL ACTIONS  
[Host]

STATE  
Action can be accepted and executed immediately.

| Name                                                                | Type      |
|---------------------------------------------------------------------|-----------|
| [HA]NodeRole::master [Kubernetes-DC11KubeTarget]                    | Placement |
| Flexera: Microsoft Windows Server 2012 Datacenter restrict to hosts | Placement |

The policies that Turbonomic generates affect Turbonomic actions in the following ways:

| Entity:         | Policy Impact on Actions:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual Machine | <ul style="list-style-type: none"> <li>Resize vCPU<br/>Limit Resize actions to stay above a specified minimum.<br/>Turbonomic will not recommend a resize that falls below a specified minimum. However, if a VM is already below the specified minimum and shows no performance problems, it will not recommend a resize up.</li> <li>Move<br/>Turbonomic restricts licensed VMs to their matching licensed Hosts. If the host has capacity, it can move unlicensed VMs onto a licensed host. However, if the host sees bottlenecks, Turbonomic will move unlicensed VMs off of the host first.<br/>Move actions can also consolidate license consumption on hosts. Assume you have one host that supports two licenses (LA and LB), and another host that only supports one license (LA). Also assume a mix of workloads, some that require both LA and LB, and some that only require LA. Move actions can consolidate the LA workloads on the LA host, leaving more resources for workloads that also require LB.</li> </ul> |
| Host            | <ul style="list-style-type: none"> <li>Provision Host<br/>Turbonomic can recommend to provision a new licensed host. For example, assume your environment includes hosts that support two different licenses (LA and LB). Now assume</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Entity: | Policy Impact on Actions:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <p>you add more workloads that require only LB licenses. Turbonomic can recommend that you provision a new host that supports the LB license.</p> <ul style="list-style-type: none"> <li>• Reconfigure Host</li> </ul> <p>Turbonomic can recommend Host Reconfigure actions to change the licensing on a host. Assume Turbonomic has executed actions to consolidate LB-only workloads on a host with LA and LB licenses. Turbonomic can recommend that you reconfigure that host to remove LA, and only support the LB license.</p> |

## ServiceNow

You can configure Turbonomic policies that log Turbonomic actions in your ServiceNow instance, and that submit actions for approval in ServiceNow workflows.

For more information concerning orchestration, see [Action Orchestration](#).

### Note:

When creating the action orchestration policy as explained in the section above, the scope of the policy must match the scope of the ServiceNow target.

## Prerequisites

- A ServiceNow user with the `web_service_admin` role and the custom role `x_turbo_turbonomic.user` that is created during installation that can communicate with Turbonomic via the REST API.

## Adding ServiceNow Targets

To add ServiceNow targets, select the **Orchestration > ServiceNow** option on the Target Configuration page and provide the following information:

- Address
  - Hostname of the ServiceNow instance without the `http` or `https` protocols. For example, `dev-env-266.service-now.com`.
- Username
  - Username for the account Turbonomic will use to connect to the ServiceNow instance
- Password
  - Password for the account Turbonomic will use to connect to the ServiceNow instance
- Client ID
  - The Client ID Turbonomic will use if Use `oAuth` is checked
- Client Secret
  - The password Turbonomic will use if Use `oAuth` is checked
- Port
  - Port used to access the ServiceNow Instance
- Use `oAuth`
  - When checked, Turbonomic will use `oAuth` authentication to connect to the ServiceNow target
- Proxy Host
  - IP address of the proxy server
- Proxy Port

Port used to access the proxy

- Proxy User

Username for the account Turbonomic will use to connect to the proxy

- Port

Port used to access the ServiceNow Instance

## ServiceNow Integration

In order to complete target addition, see the [Turbonomic Actions for ServiceNow](#) documentation.

## Private Cloud

---

A private cloud manages resources in pools to support multi-tenancy and self-service provisioning of virtual workloads. Turbonomic manages these resource pools in real time as demand fluctuates. This includes demand across resource pools, virtual datacenters (VDCs), and tenants.

Adding a Private Cloud target to Turbonomic aids in the goal of Application Resource Management (ARM) by preventing bottlenecks within and between your managed entities, resizing virtual machines and containers, prioritizing managed workloads, and reducing storage latency throughout your private cloud, ensuring that applications running in your private cloud always have the resources they need to perform, when they need to perform.

On the private cloud, you can use Turbonomic to:

- Set up charge-back and show-back for private cloud or service-provider scenarios
- For service-providers, set up scoped views to limit exposure to the customer base
- Plan hardware requirements — the planning scenarios takes cloud architectures into account

## Supply Chain

For private clouds, Turbonomic discovers resource partitions that are managed by the cloud manager, the workloads running on these partitions (the VMs and applications), and the supply that hosts workload (the physical machines and storage). Turbonomic represents these partitions as the following types of Virtual Datacenters (VDCs):

- Provider VDC

A collection of physical resources (PMs and datastores) within a private cloud. The cloud administrator has access to these resources, and defines the datacenter members. Administrators allocate Provider VDCs to manage resources that will be allocated to external customers through one or more Consumer VDCs.

- Consumer VDC

A collection of resources that are available for customers to perform self-service management of workload through the cloud. It is an environment customers can use to store, deploy, and operate virtual systems. Consumer VDCs use the resources supplied by a Provider VDC.

### Note:

Different targets use different names to refer to Virtual Data centers. In the Turbonomic supply chain, these entities are all represented by Consumer and Provider VDCs, as follows:

| <b>Turbonomic</b>   | <b>vCenter Server</b> | <b>VMM</b>            |
|---------------------|-----------------------|-----------------------|
| <b>Consumer VDC</b> | Resource Pool (Child) | Tenant or TenantQuota |
| <b>Provider VDC</b> | Resource Pool (Root)  | Cloud                 |

## Actions

Turbonomic recommends actions for private cloud infrastructures as follows:

- Provider Virtual Datacenters
  - Turbonomic does not recommend actions to perform on a Provider VDC. Instead, it recommends actions to perform on the devices running in the datacenter.
- Consumer Virtual Datacenters
  - Resize Consumer vDC
  - Provision Consumer vDC

## Monitored Resources

Turbonomic monitors the following private cloud infrastructure resources:

- Provider Virtual Datacenter
  - Memory (Mem)  
The utilization of the Datacenter's memory reserved or in use.
  - CPU  
The utilization of the Datacenter's CPU reserved or in use.
  - Storage  
The utilization of the storage attached to the Provider vDC.
- Consumer Virtual Datacenter
  - Memory (Mem)  
The utilization of the Datacenter's memory reserved or in use.
  - CPU  
The utilization of the Datacenter's CPU reserved or in use.
  - Storage  
The utilization of the storage attached to the Consumer vDC.

## Virtual Machine Manager

In a Virtual Machine Manager(VMM) environment, the VMM management server processes commands and controls communications with the Hyper-V hosts. To manage VMM, you set the management server as a target. Turbonomic communicates with that target, and also with the Hyper-V hosts that the VMM server manages. For this reason, if you add a VMM target, you should not also add individual Hyper-V hosts as targets. You must grant Turbonomic access to the VMM management server, and also to all the associated Hyper-V machines.

## Prerequisites

- VMM Dynamic Optimization disabled (recommended) or set to Low aggressiveness, with a frequency of at least 60 minutes.
- Configure remote management on the VMM management Hyper-V server. Refer to [Enabling Windows Remote Management](#)
- Grant access on every host in the cluster that Turbonomic will manage to the following WMI namespaces and subspaces:
  - root/virtualization
  - root/virtualization/v2
  - root/microsoft/windows/storage

- root/microsoft/windows/smb
- root/wmi
- root/cimv2
- root/mscluster
- Apply necessary hot fixes on the VMM host
 

For example, if you are running VMM Server on a Windows Server version earlier than Windows Server 2012 R2, you must apply the hot-fix referenced in the Microsoft Knowledge Base article #2842230 (<http://support.microsoft.com/kb/2842230>).
- Apply time synchronization
 

The VM that hosts Turbonomic must be synchronized with each target VMM management server. For instructions on synchronizing the clock on the Turbonomic server, see [Synchronizing Time](#).
- PowerShell execution must be enabled on the VMM management server.
- Configure port access
 

WinRM uses ports 5985 and 5986 for standard and secure communications, respectively. The firewall on your VMM server must open these ports.
- Preexisting Hyper-V Targets
 

If any of the hosts that make up the VMM target were added separately as Hyper-V targets (and you do not plan to exclude these hosts), these targets must be deleted. Failure to do so will create duplicate entities in the market, which will negatively impact Turbonomic performance.

## Adding VMM Targets

Turbonomic uses the address and credentials you provide to discover the VMM target. From the VMM target, Turbonomic gets the list of managed Hyper-V instances. It then uses that list to discover each Hyper-V instance. The Hyper-V credentials you provide must be valid for all of these machines.

Turbonomic will also import your Availability Sets, representing them as placement policies for the affected infrastructure.

To add VMM targets, select the **Private Cloud > VMM** option on the Target Configuration page, and provide the following information:

- The IP address or host name of the VMM management server
- Which port to use for the WSMAN connection

For a standard connection (HTTP) use 5985. For a secure connection (HTTPS) use 5986.

- Enable or disable a secure connection

If you enable a secure connection, then you must configure a certificate, and you must configure Turbonomic to communicate over HTTPS. For more information, see [Secure Setup for WSMAN](#).

Note that setting a secure connection for VMM does not also set secure connections for the underlying Hyper-V hosts. Any communications between Turbonomic and VMM will be secure. To configure secure connections to the underlying Hyper-V hosts, you must specify secure connections on each one.

- Full domain name for the user account

This domain name identifies the user account for Active Directory authentication.

- Login credentials for the Hyper-V servers that are managed by the VMM target (Optional)

Turbonomic must log into the Hyper-V servers that the VMM server manages. If you leave the Hyper-V credentials blank, then it will use the same credentials that VMM uses. If you provide Hyper-V credentials, then it will use that service account to log into every Hyper-V managed by the VMM.

Note that the service account Turbonomic uses to log into a Hyper-V host must satisfy certain requirements. For more information, see [Creating a Service User Account in Hyper-V](#).

- Hosts in the VMM target that you would like to exclude from Turbonomic.

Turbonomic will not discover or manage any hosts you enter in this field. These hosts do not participate in the market, and Turbonomic does not generate actions that involve these hosts. Enter multiple hosts in a comma-separated list. This field accepts both host names, and FQDNs (Fully Qualified Domain Names), and allows the use of the wild-card characters \* and ?.

## Microsoft VMM and Hyper-V VLANs

Windows Server Hyper-V provides support for VLANs on host and VM partitions. If your Hyper-V environment makes use of this VLAN support, then your VM moves must be sensitive to which hosts provide networking access to your defined VM networks. If a VM is a member of a given VM network, then any move of that VM must be to a host that has access to the same network.

For Hyper-V targets in a VMM environment, the Cloud Control Module is aware of the VM networks, and ensures that a move is to a host that provides connectivity over the given VM network. For example, if Host 1 and 2 provides connectivity to VM Network A, and Host 3 and 4 provide connectivity to VM Network B, Turbonomic will never recommend a VM with access to Network A, residing on Host 1, to move to Host 3. That would render the VM unable to communicate on Network A.

## Configuring SMB 3.0 File Shares Discovery

With VMM, Turbonomic can discover SMB 3.0 shares as datastores, assuming these shares have been properly added to your VMM service center. When you add shares to your VMM environment, be sure to:

- Use the Fully Qualified Domain Name of the file server

As you add the associated file server to your VMM environment (via the Add Storage Devices Wizard), be sure to specify the FQDN of the file server on the Specify Discovery Scope page of the wizard. Do not use the file server's IP address.

- Ensure that file server names are unique

Do not specify file servers with the same name, even if they belong to different domains. Turbonomic requires the file server names to be unique.

For information about setting up SMB 3.0 shares, please refer to your Microsoft documentation. For example, see [“How to Assign SMB 3.0 File Shares to Hyper-V Hosts and Clusters in VMM”](#).

## Actions

Turbonomic recommends actions for the cloud target supply chain as follows.

- Virtual Machines
  - Provision additional resources (VMem, VCPU)
  - Move Virtual Machine
  - Move Virtual Machine Storage
  - Reconfigure Storage
  - Reconfigure Virtual Machine
  - Suspend VM
  - Provision VM
- Host
  - Start
  - Provision
  - Suspend
- Consumer Virtual Datacenters
  - Resize Consumer vDC
  - Provision Consumer vDC

## Monitored Resources

Turbonomic monitors the following resources for the cloud target supply chain:

- Virtual Machine
  - – Virtual Memory
    - Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU
    - Virtual CPU is the measurement of CPU utilized by the entity.
  - Virtual Storage (VStorage)
    - Virtual Storage is the measurement storage utilized by the entity.
  - Storage Access
    - Storage Access is the measurement of IOPS utilized by the entity.
  - Latency
    - Latency is the measurement of storage latency utilized by the entity.
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters
  - Net
    - The utilization of data through the host's network adapters
  - Swap
    - The utilization of the host's swap space
  - Balloon
    - The utilization of shared memory among VMs running on the host. ESX-only
  - CPU Ready
    - The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only
- Host
  - Memory (Mem)
    - The utilization of the host's memory reserved or in use
  - CPU
    - The utilization of the host's CPU reserved or in use
  - IO
    - The utilization of the host's IO adapters
  - Net
    - The utilization of data through the host's network adapters
  - Swap
    - The utilization of the host's swap space
  - Balloon



The utilization of shared memory among VMs running on the host. ESX-only

- CPU Ready

The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only

- Datacenter

**Note:**

For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the hosts in the datacenter.

- Memory (Mem)

The utilization of the host's memory reserved or in use.

- CPU

The utilization of the host's CPU reserved or in use.

- IO

The utilization of the host's IO adapters.

- Net

The utilization of data through the host's network adapters.

- Swap

The utilization of the host's swap space.

- Balloon

The utilization of shared of memory among VMs running on the host. ESX-only.

- CPU Ready

The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only .

- Datacenter

**Note:**

For datacenter entities, Turbonomic does not monitor resources directly from the datacenter, but from the hosts in the datacenter.

- Memory (Mem)

The utilization of the host's memory reserved or in use.

- CPU

The utilization of the host's CPU reserved or in use.

- IO

The utilization of the host's IO adapters.

- Net

The utilization of data through the host's network adapters.

- Swap

The utilization of the host's swap space.

- Balloon

The utilization of shared of memory among VMs running on the host. ESX-only.

- CPU Ready

The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only .

- Provider Virtual Datacenter
  - Memory (Mem)  
The utilization of the Datacenter's memory reserved or in use.
  - CPU  
The utilization of the Datacenter's CPU reserved or in use.
  - Storage  
The utilization of the storage attached to the Provider vDC.
- Consumer Virtual Datacenter
  - Memory (Mem)  
The utilization of the Datacenter's memory reserved or in use.
  - CPU  
The utilization of the Datacenter's CPU reserved or in use.
  - Storage  
The utilization of the storage attached to the Consumer vDC.

## Public Cloud

---

The public cloud provides compute, storage, and other resources on demand. Turbonomic can analyze the performance of workloads running on the public cloud, and scale workloads as demand requires.

With public cloud targets, you can use Turbonomic to:

- Scale workloads
- Purchase discounts
- Delete unattached volumes

## Amazon Web Services

Amazon Web Services (AWS) provides a reliable and scalable infrastructure platform in the cloud. You gain access to this infrastructure through a subscription account with the appropriate organization API permissions. To specify an AWS target, you provide the credentials for that account and Turbonomic discovers the resources available to you through that account.

### Before Adding Targets

Review the following items before adding AWS targets.

- **Permissions**

The account for an AWS target must have permissions that enable Turbonomic to monitor entities in your environment, recommend actions, and execute actions.

For details, see [AWS Permissions](#).

- **Supported Regions**

Turbonomic supports discovery and management of entities in certain AWS regions, including [AWS GovCloud \(US\)](#) regions.

For details, see [Supported AWS Regions](#) and [Support for AWS GovCloud](#).

- **AWS Billing Families**

As you configure AWS targets, Turbonomic discovers AWS accounts that are consolidated into *billing families*. By recognizing billing families, Turbonomic more accurately calculates cloud investments and savings, and makes more accurate recommendations for RI coverage.

For details, see [AWS Billing Families](#).

- **Cost and Usage Report**

In order for Turbonomic to display month-to-day spend, you must create a cost and usage report in AWS and store it in an S3 bucket.

For details, see:

- [Creating Cost and Usage Reports](#) in the AWS documentation
- [Setting up an Amazon S3 bucket for Cost and Usage Reports](#) in the AWS documentation
- [Displaying AWS Spend In Turbonomic](#)

- **Reserved Instances (RIs)**

In order to discover RI utilization, you must provide Turbonomic with access to the S3 bucket that contains the AWS Cost and Usage report. Without this access, Turbonomic's purchase and scale decisions will be made without consideration of this data.

In order to discover all RIs in an AWS Billing family, you must add a billing target via the master account. This master account needs cost explorer API access. Without the master account, Turbonomic will discover only those RIs that are purchased by accounts which have been added as targets.

- **Collection of Memory Metrics**

We highly recommend enabling collection of memory metrics in your environment. Enabling memory metrics allows Turbonomic to generate Scale actions to optimize VM memory usage. For Turbonomic to collect memory metrics, you must enable the collection of these metrics on the VMs in your environment.

For details, see [AWS Memory Metrics Collection](#).

- **Firewall and Proxy Access**

If you run Turbonomic behind a proxy or firewall, then you must configure Turbonomic to allow unrestricted access to certain URLs. This enables the full set of features for AWS targets.

For details, see [Firewall and Proxy Access for AWS](#).

## Adding AWS Targets

For Turbonomic to manage an AWS account, you provide the credentials via the access key or IAM role that you use to access that account. For information about getting an Access Key for an AWS account, see the Amazon Web Services documentation.

The AWS Target has different target addition requirements based on connection via access key or IAM role.

### To add an AWS target *without* an IAM role, specify the following:

- Custom Target Name

The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

- GovCloud toggle

When enabled, this target will be added as an [AWS GovCloud \(US\)](#) target.

- Access Key

Provide the **Access Key** for the account you want to manage.

- Secret Access Key

Provide the **Access Key Secret** for the account you want to manage.

- Proxy Host

The IP of your Proxy Host

- Proxy Port

The port required for the proxy above

- Proxy User

The username required for the proxy above

- Proxy Password

The password required for the proxy above

**To add an AWS target *with* an IAM role, specify the following:**

**Note:**

Turbonomic does not support adding AWS targets via IAM roles for an instance that was deployed on-prem as an OVA or VHD image. If your instance was deployed in this way, you must add the target *without* using IAM Roles (see above).

Turbonomic deployed in Kubernetes (Red Hat OpenShift, Amazon EKS, GKE) supports adding AWS targets via IAM roles if the cluster configurations support an OIDC provider and webhook method. See [AWS Target IAM Role Requirements](#).

The Turbonomic SaaS offering also supports adding AWS targets via IAM roles. See [Turbonomic SaaS IAM Role Setup](#).

We recommend that you set up IAM access via an IAM group that has the necessary permissions. After you create this group, create a user that is a member of it and specify that user for Turbonomic to access your AWS environment. For more information about IAM Groups, see <http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html>.

- Custom Target Name

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in AWS.

- IAM Role ARN

Provide the **ARN** for the IAM role used to access the AWS Billing information.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the internet via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username for the account to log into the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

## Adding an AWS Billing Target

The Turbonomic AWS Billing target allows users to grant access to a bill which is used to discover billing family relationships. It does not provide access to any operational concern of an AWS account. Note that you can have one AWS Billing target per Turbonomic instance.

For details, see [AWS Billing](#).

## Entity Mapping

After validating your AWS targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between AWS and Turbonomic.

| <b>AWS</b>                        | <b>Turbonomic</b>    |
|-----------------------------------|----------------------|
| Virtual Machine (VM)              | Virtual Machine (VM) |
| Relational Database Service (RDS) | Database Server      |
| Storage                           | Volume               |
| Zone                              | Zone                 |
| Region                            | Region               |

## Monitored Resources

Turbonomic monitors the following resources:

- **Virtual Machine**

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- I/O Throughput

I/O Throughput is the measurement of an entity's throughput to the underlying storage.

- Net Throughput

Net Throughput is the rate of message delivery over a port.

- **Database Server**

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Connection

Connection is the measurement of database connections utilized by applications.

- **Volume**

- Storage Amount

Storage Amount is the storage capacity (disk size) of a volume.

Turbonomic discovers Storage Amount, but does not monitor utilization.

For a Kubeturbo (container) deployment that includes volumes, Kubeturbo monitors Storage Amount utilization for the volumes. You can view utilization information in the Capacity and Usage chart.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the measurement of IOPS capacity that is in use.

- I/O Throughput

I/O Throughput is the measurement of I/O throughput capacity that is in use.

## Actions

Turbonomic can recommend the following actions:

### Note:

Use the Potential Savings and Necessary Investments charts to view pending actions and evaluate their impact on your cloud expenditure.

#### • Virtual Machine

- **Stop** and **Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing RI coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

If you need more capacity, then Turbonomic will recommend actions to purchase additional RIs.

For details, see [Actions for AWS VMs](#).

#### • Database Server

- **Stop** and **Start** (also known as 'parking' actions)

Stop a Database Server for a given period of time to reduce your cloud expenses, and then start it at a later time.

- **Scale**

Scale compute and storage resources to optimize performance and costs.

For details, see [Cloud Database Server Actions](#) and [Parking: Stop or Start Cloud Resources](#).

#### • Volume

- **Scale**

Scale attached volumes to optimize performance and costs.

- **Delete**

Delete unattached volumes as a cost-saving measure. Turbonomic generates an action immediately after discovering an unattached volume.

For details, see [Cloud Volume Actions](#).

## AWS Permissions

The account for an AWS target must have permissions that enable Turbonomic to monitor entities in your environment, recommend actions, and execute actions.

### Generic AWS Permissions

You can use generic AWS permissions to set up access for the account Turbonomic uses to access your target. This is an easy way to configure a target account, but you do not have full control over the access you grant to that account. For finer control, you should set the minimum permissions.

| Permission Level                           | Required Permissions                                                                                                                                                                                                                                  |
|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Read-Only (monitoring and recommendations) | <ul style="list-style-type: none"><li>• AmazonEC2ReadOnlyAccess</li><li>• AmazonS3ReadOnlyAccess</li><li>• AmazonRDSReadOnlyAccess</li><li>• AWSConfigRoleForOrganizations (only required for consolidated billing with the master account)</li></ul> |
| Execute Actions                            | <ul style="list-style-type: none"><li>• AmazonEC2FullAccess</li><li>• AmazonS3ReadOnlyAccess</li><li>• AmazonRDSFullAccess</li><li>• AWSConfigRoleForOrganizations (only required for consolidated billing with the master account)</li></ul>         |

### Minimum Permissions

To explicitly control the access that you grant to Turbonomic, as a minimum the account for an AWS target must have the following permissions:

| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"><li>• autoscaling:DescribeAutoScalingGroups</li><li>• cloudwatch:GetMetricData</li><li>• cloudwatch:GetMetricStatistics</li><li>• cloudwatch:ListMetrics</li><li>• ec2:DescribeSpotInstanceRequests</li><li>• ec2:DescribeAvailabilityZones</li><li>• ec2:DescribeInstances</li><li>• ec2:DescribeImages</li><li>• ec2:DescribeVolumes</li><li>• ec2:DescribeVolumeStatus</li><li>• ec2:DescribeVolumesModifications</li><li>• ec2:DescribeAddresses</li><li>• ec2:DescribeRegions</li><li>• ec2:DescribeReservedInstances</li><li>• ec2:DescribeReservedInstancesModifications</li><li>• ec2:DescribeSpotPriceHistory</li><li>• ec2:DescribeAccountAttributes</li></ul> |

| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | <ul style="list-style-type: none"> <li>• ec2:ListTagsForResource</li> <li>• elasticloadbalancing:DescribeTargetGroups</li> <li>• elasticloadbalancing:DescribeTargetHealth</li> <li>• elasticloadbalancing:DescribeLoadBalancers</li> <li>• elasticloadbalancing:DescribeInstanceHealth</li> <li>• iam:GetUser</li> <li>• organizations:DescribeOrganization</li> <li>• organizations:ListAccounts</li> <li>• pi:GetResourceMetrics</li> <li>• rds:DescribeDBInstances</li> <li>• rds:DescribeDBClusters</li> <li>• rds:DescribeDBParameters</li> <li>• rds:ListTagsForResource</li> <li>• rds:DescribeOrderableDBInstanceOptions</li> <li>• servicecatalog:SearchProducts</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Action Execution         | <ul style="list-style-type: none"> <li>• autoscaling:SuspendProcesses</li> <li>• autoscaling:ResumeProcesses</li> <li>• autoscaling:DescribeLaunchConfigurations</li> <li>• autoscaling&gt;CreateLaunchConfiguration</li> <li>• autoscaling&gt;DeleteLaunchConfiguration</li> <li>• autoscaling:UpdateAutoScalingGroup</li> <li>• ec2:DescribeInstances</li> <li>• ec2:DescribeVolumes</li> <li>• ec2:DescribeVolumesModifications</li> <li>• ec2:ModifyInstanceAttribute</li> <li>• ec2:StopInstances</li> <li>• ec2:StartInstances</li> <li>• ec2:ModifyVolume</li> <li>• ec2:DescribeInstanceStatus</li> <li>• ec2:DescribeReservedInstancesOfferings</li> <li>• ec2&gt;DeleteVolume</li> <li>• iam:PassRole (required if IAM Instance Profile is specified for the ASG Launch Configuration for the ASG VM being resized)</li> <li>• KMS:CreateGrant (required if your EC2 instances use encrypted EBS volumes)</li> <li>• rds:ModifyDBInstance</li> <li>• rds:StartDBInstance</li> <li>• rds:StopDBInstance</li> <li>• servicecatalog:DescribeProduct</li> <li>• servicecatalog:ProvisionProduct</li> <li>• servicecatalog:DescribeRecord</li> </ul> |



| Turbonomic Functionality | Required Permissions                                                                  |
|--------------------------|---------------------------------------------------------------------------------------|
|                          | <ul style="list-style-type: none"> <li>• servicecatalog:ListLaunchPaths</li> </ul>    |
| Savings Plans            | <ul style="list-style-type: none"> <li>• savingsplans:DescribeSavingsPlans</li> </ul> |

## AWS Billing Target Permissions

To enable access to an AWS Billing target, the account must also include these permissions.

| Turbonomic Functionality | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"> <li>• ce:GetReservationUtilization</li> <li>• ce:GetSavingsPlansUtilizationDetails</li> <li>• ce:GetSavingsPlansUtilization</li> <li>• ce:GetSavingsPlansCoverage</li> <li>• iam:GetUser</li> <li>• organizations:DescribeOrganization</li> <li>• organizations:ListAccounts</li> <li>• pricing:DescribeServices</li> <li>• pricing:GetAttributeValues</li> <li>• s3:GetBucketAcl</li> <li>• s3:GetObject</li> <li>• sts:AssumeRole</li> <li>• sts:AssumeRoleWithWebIdentity</li> </ul> |

For details, see [AWS Billing Targets](#).

## Supported AWS Regions

Turbonomic supports discovery and management of entities in the following AWS regions:

| Region Code    | Region Name              | Notes                              |
|----------------|--------------------------|------------------------------------|
| af-south-1     | Africa (Cape Town)       | Requires opt-in within AWS console |
| ap-south-1     | Asia Pacific (Mumbai)    |                                    |
| ap-northeast-1 | Asia Pacific (Tokyo)     |                                    |
| ap-northeast-2 | Asia Pacific (Seoul)     |                                    |
| ap-northeast-3 | Asia Pacific (Osaka)     |                                    |
| ap-southeast-1 | Asia Pacific (Singapore) |                                    |
| ap-southeast-2 | Asia Pacific (Sydney)    |                                    |
| ap-southeast-3 | Asia Pacific (Jakarta)   |                                    |
| ap-east-1      | Asia Pacific (Hong Kong) | Requires opt-in within AWS console |

| Region Code   | Region Name               | Notes                              |
|---------------|---------------------------|------------------------------------|
| ca-central-1  | Canada (Central)          |                                    |
| eu-central-1  | Europe (Frankfurt)        |                                    |
| eu-south-1    | Europe (Milan)            | Requires opt-in within AWS console |
| eu-west-1     | Europe (Ireland)          |                                    |
| eu-west-2     | Europe (London)           |                                    |
| eu-west-3     | Europe (Paris)            |                                    |
| eu-north-1    | Europe (Stockholm)        |                                    |
| me-south-1    | Middle East (Bahrain)     | Requires opt-in within AWS console |
| sa-east-1     | South America (São Paulo) |                                    |
| us-east-1     | US East (N. Virginia)     |                                    |
| us-east-2     | US East (Ohio)            |                                    |
| us-west-1     | US West (N. California)   |                                    |
| us-west-2     | US West (Oregon)          |                                    |
| us-gov-east-1 | AWS GovCloud (US-East)    | Restricted Access (GovCloud)       |
| us-gov-west-1 | AWS GovCloud (US-West)    | Restricted Access (GovCloud)       |

## Unreachable AWS Regions

When Turbonomic fails to reach one or more AWS regions as part of discovering your AWS environment, then AWS discovery will fail for that target.

There may be policy decisions that prevent Turbonomic from reaching all AWS regions. For example, if you operate Turbonomic behind a firewall, you might not be able to reach all the regions that are available to your AWS account. In that case, you need to specify which regions you want Turbonomic to discover.

For information about how to specify the regions that you want Turbonomic to discover, contact your support representative.

## Support for AWS GovCloud

AWS GovCloud (US) provides dedicated regions for US government customers and their partners to architect secure cloud solutions and meet regulatory and compliance requirements.

Turbonomic discovers workloads in these regions when you add the required accounts as targets. Discovered workloads include VMs (including auto-scaling groups), volumes, database servers, and spot instances. Turbonomic recommends actions on VMs, volumes, and database servers to address performance issues and optimize costs.

## Prerequisites

When adding GovCloud targets, ensure the following prerequisites are met:

- The GovCloud account must be either a master account with organization access, or a member account belonging to a master account already added to Turbonomic. In addition, each account within the organization must have a unique name.

When you add accounts, be sure to enable the **GovCloud** toggle to indicate that the account you are adding is for GovCloud.

For details about adding accounts as targets, see [Amazon Web Services](#).

- RI utilization information is identical to non-government targets.
- A billing target with organization access containing billing information for the GovCloud target must be added to Turbonomic in order to discover RIs.

For details about billing targets, see [AWS Billing](#).

Optionally, you can add the AWS standard account that manages invoicing for a GovCloud account. When you add this standard account, it appears in the Top Accounts chart and shows the total projected billed cost for the GovCloud account and the standard account itself.

## Information in Charts

Use the following charts to view information about your government accounts and workloads.

- **Top Accounts** chart

Use the Top Accounts chart as a starting point.

| Name                                                        | Worklo... | Potential Savings ⓘ | Actions    |
|-------------------------------------------------------------|-----------|---------------------|------------|
| Gov Pay-As-You-Go<br>Azure   US Government   ██████████     | 1         | \$0.96/mo           | 1 ACTION   |
| GovEA - Development 2<br>Azure   US Government   ██████████ | 1         | \$1.11/mo           | 1 ACTION   |
| ★ Development<br>AWS   GovCloud (US)   ██████████           | 9         | \$14/mo             | 4 ACTIONS  |
| Quality Engineering<br>AWS   GovCloud (US)   ██████████     | 16        | \$64/mo             | 17 ACTIONS |
| EA - ParkMyCloud<br>Azure   Global   ██████████             | 8         | \$132/mo            | 5 ACTIONS  |

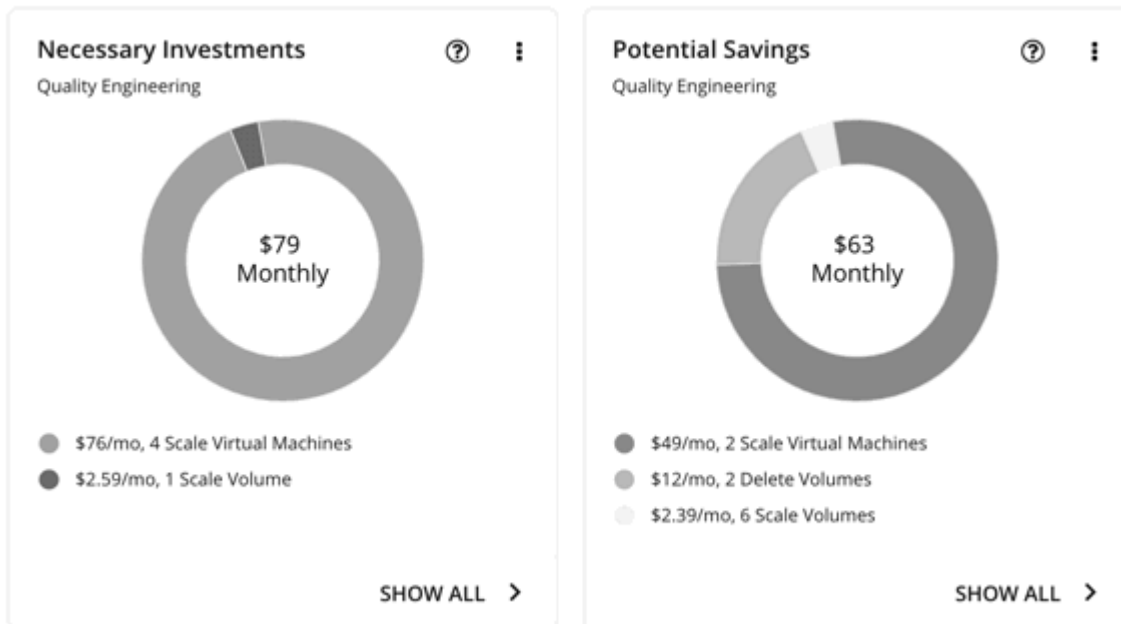
SHOW ALL >

The chart shows the GovCloud master and member accounts that you have added as targets. Accounts with a star symbol are master accounts.

Be aware that the 30-day cost is shown as **N/A** since invoicing for that account is managed through an associated AWS *standard account*. Adding this standard account as a target is optional. When added, it appears in the Top Accounts chart and shows the total billed cost for the GovCloud account and the standard account itself.

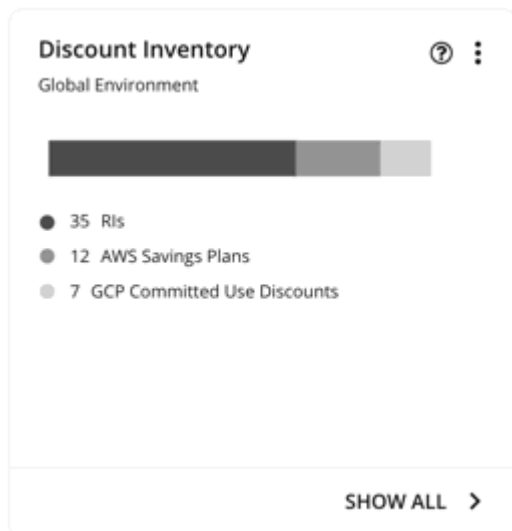
- **Necessary Investments** and **Potential Savings** charts

Set the scope to a government account or subscription, and then see the Necessary Investments and Potential Savings charts to evaluate the costs you would incur or save if you execute all the pending actions for your government workloads.



- **Discount Inventory** chart

The government accounts that you added as targets enable Turbonomic to gain full insight into the discounts that you have purchased for your government workloads. Even as you selectively add secondary targets, Turbonomic remains aware of all discounts, and how they are utilized across the board. This increases the accuracy of the allocation and purchase recommendations that Turbonomic generates for your government workloads.

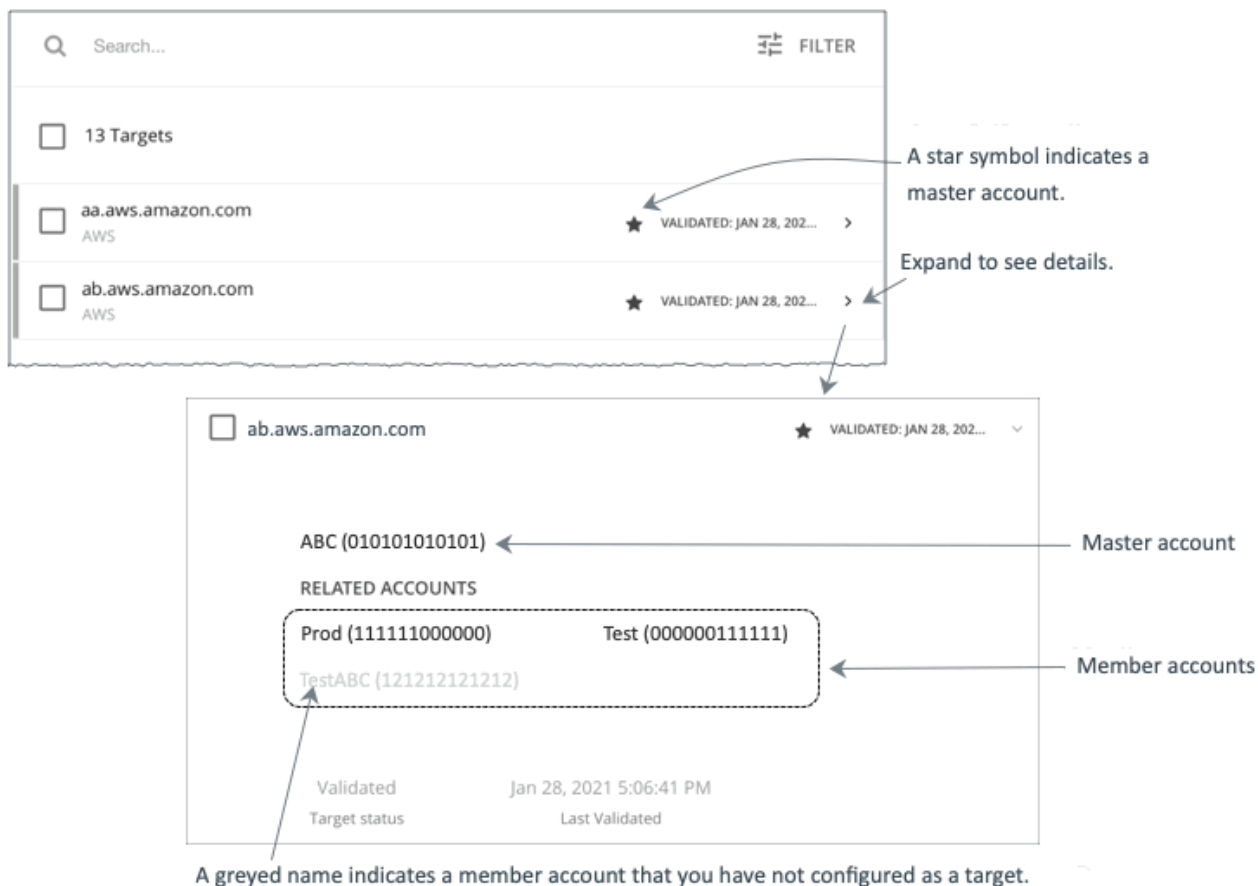


## Workload Planning

You can run an Optimize Cloud plan to identify performance and efficiency opportunities for existing government workloads, or a Migrate to Cloud plan to migrate government VM groups to another cloud provider.

For on-prem clusters, you can run a Migrate to Cloud plan to see how you can safely migrate the VMs in these clusters to a government account/subscription and region.

## AWS Billing Families



As you configure AWS targets, Turbonomic discovers AWS accounts that are consolidated into *billing families*. A billing family has one *master* account, and zero or more *member* accounts. By recognizing billing families, Turbonomic more accurately calculates cloud investments and savings, and makes more accurate recommendations for RI coverage.

In the Targets user interface, master accounts appear in bold, with a star next to them. You can expand the account entry to see the related member accounts. If you expand the entry for a member account, then the related accounts includes the family master, indicated by a star.

For RI purchases, different accounts in a billing family can share the same RI resources. At the same time, accounts in other billing families cannot use those RIs. This adds flexibility to your RI coverage, while maintaining order over the billing.

In Turbonomic, if you enable Billing Family Recognition, then you can see the billing family master and member accounts in the Targets user interface, and Turbonomic can recommend proper RI purchases within the correct billing families.

To enable Billing Family Recognition, ensure the following as you configure your AWS targets:

- Use the proper role for each AWS target

To properly discover billing family information for a target, you must give Turbonomic credentials for an AWS role that includes the permission, `organizations:DescribeOrganization`. With that permission, Turbonomic can:

- Discover master accounts and member accounts in different billing families
- Display the account names in the user interface
- Discover billing information for each family and account
- Recommend RI actions that respect billing family boundaries

- Configure targets for the complete billing family

One billing family can consolidate a number of AWS accounts. For Turbonomic to include these accounts in its analysis, you must configure each one as a separate target. If you do not configure all the accounts in a billing family, then Turbonomic cannot discover complete billing information for that family, and its analysis will be based on incomplete information.

Turbonomic displays member accounts that have been configured as targets in regular text. For members that Turbonomic discovers but have not been configured as targets, Turbonomic displays their names in grayed text.

If you have enabled Billing Family Recognition, you should keep the following points in mind:

- Billing families can grow

Turbonomic regularly checks the membership of your billing families. If it discovers a new member account, it adds that account to the list of members. If you have already configured the account as a target, then Turbonomic includes the new member in its analysis of billing families. If the new member is not already a target, then Turbonomic lists the new member in grayed text.

- You can configure discounts per billing family

Turbonomic includes a feature to set a discount for a billing group, and to override that discount for specific template families within that scope. For more information, see [Cloud Discounts](#) and [Discount Override: AWS](#).

- You might see master accounts that have no member accounts

AWS treats every account you create as a part of a billing family. Assume you created an account, but you had no reason to consolidate its billing with any other accounts. In that case, the account appears in the Turbonomic user interface as a master account, but it has no member accounts.

## AWS Memory Metrics Collection

We highly recommend enabling collection of memory metrics in your environment. Enabling memory metrics allows Turbonomic to generate Scale actions to optimize VM memory usage. For Turbonomic to collect memory metrics, you must enable the collection of these metrics on the VMs in your environment.

Some of the steps to do this are different depending on whether your VM is running a Linux or Windows OS. To enable memory statistics on your VMs, you must meet the following requirements:

- The VM image must have an SSM agent installed

- Linux VMs:

By default, Linux AMIs dated 2017.09 and later include an installed SSM Agent.

- Windows VMs:

You must install the SSM agent on the VMs. For more information, see [Working with SSM Agent](#).

- Access to the CloudWatch service

Your AWS Instance must have internet access or direct access to CloudWatch so it can push data to CloudWatch.

- Access from Turbonomic

For Turbonomic to access the memory data, the account that it uses to connect to the AWS target must include the correct permissions. If you configured the AWS target via an AWS key (not an IAM role), then you must include the permissions as specified in the section for configuring an AWS target.

If you use an IAM role for the Turbonomic connection, then that role must include the following as a minimum:

- `AmazonEC2ReadOnlyAccess`

- `AmazonS3ReadOnlyAccess`

To set up the collection of memory statistics for your VMs:

1. Attach an IAM role to each VM instance.

Each EC2 instance must have an attached IAM role that grants CloudWatch access. To grant that access, we suggest you include the AmazonSSMFullAccess policy in the role.

Use the AWS System Manager to attach the necessary roles to your VMs.

**Note:**

If you want to grant the role lesser access, you can use the AmazonEC2RoleforSSM policy. This is a custom policy that allows the action `ssm:GetParameter` to access the resource, `arn:aws:ssm:*:*:parameter/*`.

2. Install the CloudWatch agent on your Linux VMs.

Navigate to the AWS System Manager service for the account and region that you want to configure. In the service, navigate to the **Run Command** screen and set up the **AWS-ConfigureAWSPackage** command to install **AmazonCloudWatchAgent** on your VMs. For more information, please see the AWS documentation.

3. Create configuration data for the CloudWatch agent.

The configuration data is a JSON object that you will add to as a parameter to the Parameter Store. The object must include the following, depending on whether it's for a Linux or a Windows VM instance.

• **Linux Configuration:**

```
{
 "agent": {
 "metrics_collection_interval": 60,
 "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
 },
 "metrics": {
 "namespace": "custom",
 "metrics_collected": {
 "mem": {
 "measurement": [
 {
 "name": "mem_available", "rename": "MemoryAvailable", "unit": "Bytes"
 }
]
 }
 }
 },
 "append_dimensions": {
 "AutoScalingGroupName": "${aws:AutoScalingGroupName}",
 "ImageId": "${aws:ImageId}",
 "InstanceId": "${aws:InstanceId}",
 "InstanceType": "${aws:InstanceType}"
 }
}
```

• **Windows Configuration:**

```
{
 "metrics": {
 "namespace": "Windows System",
 "append_dimensions": {
 "InstanceId": "${aws:InstanceId}"
 },
 "aggregation_dimensions": [["InstanceId"]],
 "metrics_collected": {
 "Memory": {
 "measurement": [
 { "name": "Available Bytes", "rename": "MemoryAvailable", "unit": "Bytes" }
],
 "metrics_collection_interval": 60
 },
 "Paging File": {
 "measurement": [
 { "name": "% Usage", "rename": "paging_used" }
]
 }
 }
 }
}
```

```

],
 "metrics_collection_interval": 60,
 "resources": [
 "*"
]
 }
}
}
}

```

Note that you can configure optional parameters for the CW Namespace and region. However, if you configure more metrics for CloudWatch to collect, these metrics do not affect Turbonomic analysis and they do not show up in the user interface. Turbonomic only tracks Used Memory statistics.

#### 4. Create a parameter store.

##### a. Create a parameter.

In the AWS System Manager, navigate to **Parameter Store** and create a parameter. Copy and paste the JSON agent configuration (created above) into the parameter **Value** field.

##### b. Name the parameter

For example, AmazonCloudWatch-MyMemoryParam. You can use a different name, but per the Amazon documentation, the name *must* begin with AmazonCloudWatch. For more information, see [Store the CloudwatchConfig File in Parameter Store](#).

You must remember this parameter name.

##### c. Click to create the parameter.

#### 5. Deploy the CloudWatch parameter to your VMs.

In the AWS System Manager, navigate to the **Run Command** screen to configure and run the **AmazonCloudWatch-ManagedAgent** command. The configuration should include:

- **Action:** configure
- **Mode:** ec2
- **Optional Configuration Source:** ssm
- **Optional Configuration Location:** Give the name of the parameter that you created above.
- **Optional Restart:** yes (this restarts the CloudWatch Agent, not the VM instance)
- **Targets:** The VMs that you will deploy the CloudWatch configuration to

When the command is configured, run it. This configures collection of memory metrics for your instances.

#### 6. Verify that you are collecting Memory metrics for your instances.

Navigate to the CloudWatch page, and display **Metrics** in the **CWAgent** namespace. Then inspect the instances by ID to verify that you can see MemoryAvailable metrics.

For more information about enabling memory statistics for AWS, see the following Support articles:

- For Linux:

[Setting up CloudWatch Memory Metrics on Linux Instances on AWS.](#)

- For Windows:

[Setting up CloudWatch Memory Metrics on Windows Instances on AWS.](#)

## Firewall and Proxy Access for AWS

If you run Turbonomic behind a proxy or firewall, then you must configure Turbonomic to allow unrestricted access to the following URLs. This enables the full set of features for AWS targets.

### Note:

If your firewall performs IP based routing, be aware that Amazon reserves the right to change the DNS names on these endpoints over time. Also note, the AWS SDK is designed to use dynamic endpoints. It is



possible that Amazon will change the endpoints the SDK depends on over time. As a result, the names of endpoints you must access might be different than the names listed here.

- Autoscaling  
autoscaling.{region-id}.amazonaws.com
- AWS Bill  
{bucket-name}.s3.{region-containing-report}.amazonaws.com
- CloudWatch  
monitoring.{region-id}.amazonaws.com
- CloudWatch Events  
events.{region-id}.amazonaws.com
- CloudWatch Logs  
logs.{region-id}.amazonaws.com
- Cost Explorer  
ce.us-east-1.amazonaws.com
- EC2  
ec2.{region-id}.amazonaws.com
- Elastic Load Balancing  
elasticloadbalancing.{region-id}.amazonaws.com
- IAM  
iam.amazonaws.com
- Organizations  
organizations.{region-id}.amazonaws.com
- Performance Insights  
pi.{region-id}.amazonaws.com
- Price List  
pricing.us-east-1.amazonaws.com
- Relational Database Service (RDS)  
rds.{region-id}.amazonaws.com
- Resource Groups  
resource-groups.{region-id}.amazonaws.com
- Savings Plans (Optional: Required if you use Savings Plans in your environment)  
savingsplans.amazonaws.com
- Service Catalog  
servicecatalog.{region-id}.amazonaws.com
- S3  
s3.{region-id}.amazonaws.com
- Storage Gateway  
storagegateway.{region-id}.amazonaws.com

## AWS Billing

The Turbonomic AWS Billing target allows users to grant access to a bill which is used to discover billing family relationships. It does not provide access to any operational concern of an AWS account. Note that you can have one AWS Billing target per Turbonomic instance.

### Before Adding Targets

Review the following items before adding AWS Billing targets.

- **Permissions**

To enable access to an AWS Billing target, the AWS account must include certain permissions.

For details, see [AWS Billing Target Permissions](#).

- **Cost and Usage Report**

Billing targets use Cost and Usage reports.

In order for Turbonomic to display month-to-day spend, you must create a cost and usage report in AWS and store it in an S3 bucket.

For details, see:

- [Creating Cost and Usage Reports in the AWS documentation](#)
- [Setting up an Amazon S3 bucket for Cost and Usage Reports in the AWS documentation](#)
- [Displaying AWS Spend In Turbonomic](#)

- **Price Adjustments**

Cloud service providers can offer their own price lists, including special costs for services or discounts for workloads. However, Turbonomic does not discover these adjustments. For example, to reflect any discounted prices in the Turbonomic display and in Turbonomic analysis, you must manually configure those discounts. In Turbonomic, you configure such discounts via **Price Adjustments** for specific billing groups in your cloud environment.

Price Adjustments are needed for AWS to show proper discounted rates that customers have, even after adding the AWS Master Billing Account target. If Price Adjustments are not set, Turbonomic will show on-demand pricing, which results in incorrect cost numbers in actions and the UI. See [Price Adjustments](#) for more information.

### Adding an AWS Billing Target

The AWS Billing target has different target addition requirements based on connection via key or IAM role.

#### To add an AWS Billing *without* an IAM role, specify the following:

- Custom Target Name

The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

- Access Key

Provide the **Access Key** for the account you want to manage.

- Secret Access Key

Provide the **Access Key Secret** for the account you want to manage.

- Cost and Usage Report Bucket

Name of the S3 bucket that contains the AWS Cost and Usage report.

- Cost and Usage Report Path

Path in the S3 bucket to the AWS Cost and Usage report.

- Cost and Usage Report Region

Region of the S3 bucket that contains the AWS Cost and Usage report.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to AWS via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username for the account to log into the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

**To add an AWS Billing target *with* an IAM Role, specify the following:**

- Custom Target Name

The display name that will be used to identify the target in the Target List. This is for display in the UI only; it does not need to match any internal name.

- IAM Role ARN

Provide the **ARN** for the IAM role used to access the AWS Billing information.

- Cost and Usage Report Bucket

Name of the S3 bucket that contains the AWS Cost and Usage report.

- Cost and Usage Report Path

Path in the S3 bucket to the AWS Cost and Usage report.

- Cost and Usage Report Region

Region of the S3 bucket that contains the AWS Cost and Usage report.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to AWS via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username for the account to log into the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

## Actions

Turbonomic does not recommend actions for AWS Billing targets. However, the billing information will be used in conjunction with the AWS target to generate actions.

## Monitored Resources

Turbonomic does not monitor resources for AWS Billing targets.

## Google Cloud Platform

Google Cloud Platform (GCP) provides a scalable infrastructure platform in the cloud. Turbonomic gains access to this infrastructure through a GCP [service account](#) with the appropriate permissions.

## Before Adding Targets

Review the following items before adding GCP service accounts as targets.

- **Service Accounts**

When you add a service account as a target, Turbonomic discovers the projects that define compute, storage, and networking resources for your workloads. It then creates a derived target for each discovered project. Derived targets are not directly modifiable within Turbonomic but can be validated like any other target.

Turbonomic discovers a broader resource hierarchy if you add a service account target with permissions to retrieve folders or your entire GCP organization.

Turbonomic uses cost data in its analysis to make accurate recommendations for your workloads. To enable cost discovery and monitoring, you must grant your GCP service account "Billing Account Viewer" access to the related [billing accounts](#), and then add those billing accounts as targets.

For instructions on creating service accounts in a `gcloud` shell session, see [GCP Target Service Account](#) and [GCP Billing Target Service Account](#).

- **Permissions**

To configure GCP targets, you create service accounts with roles that grant Turbonomic the permissions it needs to discover your GCP resources and costs, and to execute actions (optional).

For details, see [GCP Permissions](#).

- **Supported Regions and Zones**

Turbonomic supports discovery and management of workloads in all currently available GCP [regions and zones](#).

- **GCP APIs**

For Turbonomic to discover your GCP environment and billing details, you must enable certain APIs.

For details, see [Enabling Required GCP APIs](#).

- **Collection of Memory Metrics**

We highly recommend enabling collection of memory metrics in your GCP environment. With memory metrics, Turbonomic can generate actions that not only boost performance but also maximize your savings.

GCP collects these metrics via [Ops Agent](#). In order for Turbonomic to retrieve these metrics, you must install and configure Ops Agent on each VM that it monitors. See [Ops Agent installation instructions here](#), and configuration details [here](#).

**Note:**

GCP recommends using Ops Agent instead of its [legacy monitoring agent](#).

- **Firewall and Proxy Access**

If you run Turbonomic behind a firewall, you must configure Turbonomic to allow unrestricted access to the following GCP URLs:

- <https://cloudresourcemanager.googleapis.com>
- <https://cloudbilling.googleapis.com>
- <https://compute.googleapis.com>
- <https://monitoring.googleapis.com>
- <https://bigquery.googleapis.com>

## Adding GCP Targets

Once a service account has been properly configured for use with Turbonomic, you must add it as a target from the Target Configuration page.

Specify the following when adding the target:

- Display Name

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in GCP.

- Service Account Key (JSON)

The service account key for the account you want to manage. Paste the JSON object for the account key into this field.

- Proxy Information

Specify proxy information only if you connect to a GCP target via proxy.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username

The username to use with the proxy specified above.

- Proxy Password

The password to use with the proxy specified above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

## Adding GCP Billing Targets

Turbonomic uses cost data in its analysis to make accurate recommendations for your workloads. To enable cost discovery and monitoring, you must grant your GCP service account "Billing Account Viewer" access to the related billing accounts, and then add those billing accounts as targets.

For details, see GCP Billing.

## Entity Mapping

After validating your GCP targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between GCP and Turbonomic.

| GCP                           | Turbonomic                                                                                                                                                                                                                                            |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual Machine (VM) Instance | Virtual Machine (VM)<br><b>Note:</b><br>Turbonomic discovers GCP labels attached to VMs as tags. You can filter VMs by tags when you use Search or create groups. The Action Details page for a pending VM action also lists all the discovered tags. |
| Storage/Disk                  | Volume                                                                                                                                                                                                                                                |
| Zone                          | Zone                                                                                                                                                                                                                                                  |
| Region                        | Region                                                                                                                                                                                                                                                |

**Note:**

GCP *projects*, *folders*, and *billing accounts* do not appear as entities in the supply chain. Use Search to scope to these resources. In Search, projects are grouped under Accounts, folders under Folders, and billing accounts under Billing Families.

For billing accounts, the latest billing data available from Google is always a few days old. As a result, billing-related charts in Turbonomic (such as the Billed Cost chart) do not have data for the current day. In addition, billing data shown in billing-related charts and GCP Billing reports might not match because Turbonomic uses UTC, while GCP uses local time. However, costs shown in both places are correct and reliable.

## Monitored Resources

Turbonomic monitors the following resources:

### • Virtual Machine

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

In order for Turbonomic to retrieve vMem metrics, you must install and configure [Ops Agent](#) on each VM that it monitors.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

Turbonomic calculates CPU based on the normalized CPU frequency and the number of vCPUs for a given VM. Normalized CPU frequency takes into account performance variations seen in different models of a given CPU platform. Because frequency is normalized, charts might show utilization values that are slightly higher than 100% (for example, 100.03%) when capacity is fully utilized.

- Net Throughput

Net Throughput is the rate of message delivery over a port.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- I/O Throughput

I/O Throughput is the measurement of an entity's throughput to the underlying storage.

For both Storage Access (IOPS) and I/O Throughput, Turbonomic calculates capacity or uses GCP-published capacity data, depending on the VM's machine type and disk.

- Shared-core machine types share a physical core and are used for running small, non-resource intensive apps.

For shared-core machine types with *standard disks*, Turbonomic uses capacity data that GCP publishes [here](#). Note that there are no published values for e2-micro and e2-small, so Turbonomic assumes the e2-medium capacity for these machine types.

For shared-core machine types with *SSDs*, Turbonomic calculates capacity based on the observed maximum limit that can be achieved for IOPS and I/O throughput, and uses the calculated capacity to analyze utilization more accurately.

- For machine types that are *not* shared-core:

- Turbonomic uses [published](#) capacity data and assumes that I/O block size is 16KB per I/O.
- For machine types with [persistent disks](#), Turbonomic assumes that the published capacity for the *SSD* disk type also applies to the *balanced* and *extreme* disk types. When a VM is attached to at least one of these disk types, capacity is assumed to be the per-VM limit for the SSD disk

type. When a VM is attached only to the *standard* disk type, capacity is the per-VM limit for the standard disk type.

- **Volume**

Currently, Turbonomic does not monitor resources for GCP volumes. It only monitors their attachment state and then generates delete actions for unattached volumes.

## Actions

Turbonomic can recommend the following actions:

### Note:

Use the Potential Savings and Necessary Investments charts to view pending actions and evaluate their impact on your cloud expenditure.

- **Virtual Machine**

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing Committed Use Discount (CUD) coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

Actions to purchase CUDs will be introduced in a future release.

- **Stop and Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

- **Reconfigure**

GCP provides a specific set of machine types for each zone in a region. If you create a policy that restricts a VM to certain machine types and the zone it is currently on does not support all of those machine types, Turbonomic will recommend a reconfigure action as a way to notify you of the non-compliant VM.

For details, see [Actions for GCP VMs](#).

- **Volume**

### Delete

Delete unattached volumes as a cost-saving measure. Turbonomic generates an action immediately after discovering an unattached volume.

Turbonomic currently supports delete actions for [zonal](#) (single zone) persistent disks.

Points to consider:

- Delete actions for [regional](#) persistent disks will be introduced in a future release. Currently, Turbonomic discovers these volumes, but does not show actions or costs in charts.
- Turbonomic will never generate delete actions for local SSDs since they are always attached to VMs. GCP automatically deletes local SSDs when you delete the corresponding VMs.

For details, see [Cloud Volume Actions](#).

## GCP Target Service Account

This topic guides you through the steps to create a valid service account that Turbonomic can use to connect with a GCP operational target. The access you grant to this service account determine the access Turbonomic has to discover and manage entities in your GCP infrastructure. To create this account you will:

- Create a service account in GCP

- Generate key file for the service account
- Create custom roles for the project(s) you want Turbonomic to manage
- Add the custom roles to the GCP Service Account, plus the predefined Billing Account Viewer role

## Creating a Service Account in GCP

Open a `gcloud shell` session in the project that you want to host the new Service Account. In `gcloud shell`, execute the following command, where:

- `$SERVICE_ACCOUNT_ID` is the unique alphanumeric ID that you assign to this service account
- `$SERVICE_ACCOUNT_DISPLAY_NAME` is the display name that you want for the service account

```
$ gcloud iam service-accounts create $SERVICE_ACCOUNT_ID \
 --display-name="$SERVICE_ACCOUNT_DISPLAY_NAME" \
 --format=text --quiet
```

When you execute the command, the shell should display the following output:

```
displayName: <SERVICE_ACCOUNT_DISPLAY_NAME>
email: <SERVICE_ACCOUNT_EMAIL>
etag: MDEwMjE5MjA=
name: <SERVICE_ACCOUNT_NAME>
projectId: <SA_DEFAULT_PROJECT>
uniqueId: 102200949905427524050
```

Note the following fields in the output for later use:

- `email`
- `name`
- `projectId`

### Note:

The `projectId` field identifies the project that hosts this Service Account. You should note that project for future reference, in case you want to review or edit the Service Account.

## Generating the Key File

To generate the key file, execute the following command in the `gcloud shell`:

```
$ gcloud iam service-accounts keys create $SA_KEY_FILE_NAME \
 --iam-account=$SERVICE_ACCOUNT_EMAIL
```

Be sure to save the key file – you need it when you configure the GCP Target in Turbonomic.

## Creating Custom Roles

You must create two different roles for access to your GCP projects and organization, and you can create a third role for permission to execute actions in projects.

### Note:

To target specific folders, define a custom role at the organization level. It is not possible to define custom roles at the folder level.

#### 1. Project Access Role:

In a location you can access from your `gcloud shell` session, create a file named `TurbonomicSaAccessProject.yaml`. Edit the file to have the following content:

```
title: "Turbonomic Role: Min Accecss - Project"
description: "Minimal Required Permissions for Turbonomic to manage the GCP Project"
stage: "ALPHA"
includedPermissions:
discovery
```



```

- resourceManager.projects.get
- compute.regions.list
- compute.zones.list
- compute.machineTypes.list
- compute.machineTypes.get
- compute.disks.list
- compute.disks.get
- compute.diskTypes.list
- compute.instances.list
- compute.instances.get
- compute.instanceGroupManagers.list
- compute.instanceGroupManagers.get
CUD
- compute.commitments.list
Metrics Monitoring
- logging.views.list
- logging.views.get
- monitoring.services.get
- monitoring.services.list
- monitoring.timeSeries.list
- serviceusage.services.get

```

Then execute the following command to create the custom role in your organization, where you substitute `TurbonomicSaAccessProject.yaml` with the path to your yaml file:

```

$ gcloud iam roles create $CUSTOM_ROLE_NAME \
 --project=$PROJECT_ID \
 --file=TurbonomicSaAccessProject.yaml

```

In the resulting console output, note the role name. You need that name to add the role to your service account.

## 2. Organization Access Role:

In a location you can access from your `gcloud` shell session, create a file named `TurbonomicSaAccessOrg.yaml`. Edit the file to have the following content:

```

title: "Turbonomic Role: Access - Organization"
description: "Minimal Required Permissions for Turbonomic to access the GCP Organization"
stage: "ALPHA"
includedPermissions:
Organization Structure View
- resourceManager.organizations.get
- resourceManager.projects.list
- resourceManager.projects.get
- resourceManager.folders.list
- resourceManager.folders.get
- billing.resourceAssociations.list

```

Then execute the following command to create the custom role in your organization, where you substitute `TurbonomicSaAccessOrg.yaml` with the path to your yaml file:

```

$ gcloud iam roles create $CUSTOM_ROLE_NAME \
 --organization=$ORGANIZATION_ID \
 --file=TurbonomicSaAccessOrg.yaml

```

In the resulting console output, note the role name. You need that name to add the role to your service account.

## 3. Project Action Role:

Create this role if you want Turbonomic to execute actions in your GCP environment.

In a location you can access from your `gcloud` shell session, create a file named `TurbonomicSaProjectAction.yaml`. Edit the file to have the following content:

```

title: "Turbonomic Role: Project Action Execution"
description: "Grant Turbonomic to execute actions in the GCP Project"
stage: "ALPHA"
includedPermissions:
Action Execution
- compute.globalOperations.get
- compute.instances.setMachineType
- compute.instances.start

```

```
- compute.instances.stop
- compute.disks.delete
- compute.regionOperations.get
- compute.zoneOperations.get
```

Then execute the following command to create the custom role in your organization, where you substitute `TurbonomicSaProjectAction.yaml` with the path to your yaml file:

```
$ gcloud iam roles create $CUSTOM_ROLE_NAME \
 --project=$PROJECT_ID \
 --file=TurbonomicSaProjectAction.yaml
```

In the resulting console output, note the role name. You need that name to add the role to your service account.

## Adding Custom Roles to the Service Account

You will now add the three roles you created to your Service Account, plus the predefined Billing Viewer role.

1. (Required) Add the predefined Billing Viewer role.

In the GCP Console, go to **Billing > Account Management**. This page should display the billing account and project you are going to use as the GCP Billing target in Turbonomic.

Add the **Billing Account Viewer** role to the Service Account

2. (Required) Add the *Organization Access* role that you created for the Service Account.

In the `gcloud console` session, execute this command, where `$ROLE_NAME` is the role name you noted when you created the Organization Access role:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
 --member=serviceAccount:$SERVICE_ACCOUNT_EMAIL --role=$ROLE_NAME
```

3. (Required) Add the *Project Access* role that you created for the Service Account.

In the `gcloud console` session, execute this command, where `$ROLE_NAME` is the role name you noted when you created the Project Access role:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
 --member=serviceAccount:$SERVICE_ACCOUNT_EMAIL --role=$ROLE_NAME
```

4. (Optional) Add the *Project Action* role that you created for the Service Account.

You only need to add this role if you want Turbonomic to execute actions in your GCP project.

In the `gcloud console` session, execute this command, where `$ROLE_NAME` is the role name you noted when you created the Project Action role:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
 --member=serviceAccount:$SERVICE_ACCOUNT_EMAIL --role=$ROLE_NAME
```

You now have created a Service Account that you can use to configure your GCP operational target.

## GCP Billing Target Service Account

This topic guides you through the steps to create a valid service account that Turbonomic can use to connect with a GCP Billing target. To create this account you will:

- Create a service account in GCP
- Generate key file for the service account
- Create a custom role in the project(s) you want Turbonomic to manage
- Add the created custom role to the GCP Service Account, plus the predefined Billing Account Viewer role

## Creating a Service Account in GCP

Open a `gcloud` shell session in the project that you want to host the new Service Account. In `gcloud` shell, execute the following command, where:

- `$SERVICE_ACCOUNT_ID` is the unique alphanumeric ID that you assign to this service account
- `$SERVICE_ACCOUNT_DISPLAY_NAME` is the display name that you want for the service account

```
$ gcloud iam service-accounts create $SERVICE_ACCOUNT_ID \
 --display-name="$SERVICE_ACCOUNT_DISPLAY_NAME" \
 --format=text --quiet
```

When you execute the command, the shell should display the following output:

```
displayName: <SERVICE_ACCOUNT_DISPLAY_NAME>
email: <SERVICE_ACCOUNT_EMAIL>
etag: MDEwMjE5MjA=
name: <SERVICE_ACCOUNT_NAME>
projectId: <SA_DEFAULT_PROJECT>
uniqueId: 102200949905427524050
```

Note the following fields in the output for later use:

- `email`
- `name`
- `projectId`

### Note:

The `projectId` field identifies the project that hosts this Service Account. You should note that project for future reference, in case you want to review or edit the Service Account.

## Generating the Key File

To generate the key file, execute the following command in the `gcloud` shell:

```
$ gcloud iam service-accounts keys create $SA_KEY_FILE_NAME \
 --iam-account=$SERVICE_ACCOUNT_EMAIL
```

Be sure to save the key file – you need it when you configure the GCP Billing target in Turbonomic

## Creating a Custom Billing Role

In a location you can access from your `gcloud` shell session, create a file named `TurbonomicSaBilling.yaml`. Edit the file to have the following content:

```
title: "Turbonomic Billing Data Viewer Role"
description: "Minimal Required Permissions for Turbonomic to view billed cost and pricing
stored in the GCP Project"
stage: "ALPHA"
includedPermissions:
- bigquery.tables.get
- bigquery.tables.getData
- bigquery.tables.list
- bigquery.jobs.create
derived cost probe will need
- compute.regions.list
- compute.zones.list
- compute.commitments.list
- compute.diskTypes.list
- compute.machineTypes.list
```

Then execute the following command to create the custom role in your organization, where you substitute `TurbonomicSaBilling.yaml` with the path to your yaml file:

```
$ gcloud iam roles create $CUSTOM_ROLE_NAME \
 --project=$PROJECT_ID \
 --file=TurbonomicSaBilling.yaml
```

In the resulting console output, note the role name. You need that name to add the role to your service account.

## Adding the Custom Roles to the Service Account

You will now add the Custom Billing role you created to your Service Account, plus the predefined Billing Viewer role.

1. (Required) Add the predefined Billing Viewer role.

In the GCP Console, go to **Billing > Account Management**. This page should display the billing account and project you are going to use as the GCP Billing target in Turbonomic.

Add the **Billing Account Viewer** role to the Service Account

2. (Required) Add the *Custom Billing* role that you created for the Service Account.

In the `gcloud console` session, execute this command, where `$ROLE_NAME` is the role name you noted when you created the Custom Billing role:

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
 --member=serviceAccount:$SERVICE_ACCOUNT_EMAIL --role=$ROLE_NAME
```

You now have created a Service Account that you can use to configure your GCP Billing target.

## GCP Permissions

To configure GCP targets, you create service accounts with roles that grant Turbonomic the permissions it needs to discover your GCP resources and costs, and to execute actions (optional).

### Resource Discovery Permissions

The following permissions are required to discover GCP workloads and resources.

| <b>Scope</b> | <b>Minimum Authorization Scopes</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | <b>Minimum IAM Roles/Permissions</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Project      | <p>One of the following scope sets:</p> <p>Set 1:</p> <ul style="list-style-type: none"> <li>• https://www.googleapis.com/auth/cloudplatformprojects.readonly</li> <li>• https://www.googleapis.com/auth/cloud-platform</li> <li>• https://www.googleapis.com/auth/cloud-billing.readonly</li> </ul> <p>Set 2:</p> <ul style="list-style-type: none"> <li>• https://www.googleapis.com/auth/cloudplatformprojects.readonly</li> <li>• https://www.googleapis.com/auth/compute</li> <li>• https://www.googleapis.com/auth/monitoring.read</li> <li>• https://www.googleapis.com/auth/cloud-billing.readonly</li> </ul> | <ul style="list-style-type: none"> <li>• resourcemanager.projects.get</li> <li>• compute.regions.list</li> <li>• compute.zones.list</li> <li>• compute.machineTypes.list</li> <li>• compute.machineTypes.get</li> <li>• compute.disks.list</li> <li>• compute.disks.get</li> <li>• compute.diskTypes.list</li> <li>• compute.instances.list</li> <li>• compute.instances.get</li> <li>• compute.instanceGroupManagers.list</li> <li>• compute.instanceGroupManagers.get</li> <li>• logging.views.list</li> <li>• logging.views.get</li> <li>• monitoring.services.get</li> <li>• monitoring.services.list</li> <li>• monitoring.timeSeries.list</li> <li>• serviceusage.services.get</li> </ul> |
| Folder       | <p>To target specific folders, define a custom role at the organization level (see the next item). It is not possible to define custom roles at the folder level.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Scope        | Minimum Authorization Scopes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Minimum IAM Roles/Permissions                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Organization | <p>One of the following scope sets:</p> <p>Set 1:</p> <ul style="list-style-type: none"> <li>• <a href="https://www.googleapis.com/auth/cloud-platform">https://www.googleapis.com/auth/cloud-platform</a></li> <li>• <a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a></li> </ul> <p>Set 2:</p> <ul style="list-style-type: none"> <li>• <a href="https://www.googleapis.com/auth/cloud-platform.read-only">https://www.googleapis.com/auth/cloud-platform.read-only</a></li> <li>• <a href="https://www.googleapis.com/auth/iam.test">https://www.googleapis.com/auth/iam.test</a></li> <li>• <a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a></li> </ul> <p>Set 3:</p> <ul style="list-style-type: none"> <li>• <a href="https://www.googleapis.com/auth/cloudplatformorganizations.readonly">https://www.googleapis.com/auth/cloudplatformorganizations.readonly</a></li> <li>• <a href="https://www.googleapis.com/auth/cloudplatformfolders.readonly">https://www.googleapis.com/auth/cloudplatformfolders.readonly</a></li> <li>• <a href="https://www.googleapis.com/auth/cloudplatformprojects.readonly">https://www.googleapis.com/auth/cloudplatformprojects.readonly</a></li> <li>• <a href="https://www.googleapis.com/auth/iam.test">https://www.googleapis.com/auth/iam.test</a></li> <li>• <a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a></li> </ul> | <ul style="list-style-type: none"> <li>• All project-level permissions</li> <li>• <code>resourcemanager.organizations.get</code></li> <li>• <code>resourcemanager.folders.get</code></li> <li>• <code>resourcemanager.folders.list</code></li> <li>• <code>resourcemanager.projects.list</code></li> <li>• <code>resourcemanager.projects.get</code></li> <li>• <code>billing.resourceAssociations.list</code></li> </ul> |

### Cost Discovery Permissions

The following permissions are required to discover GCP billing families, billed costs, negotiated pricing, and committed use discounts.

| Minimum Authorization Scopes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Minimum IAM Roles/Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• All <a href="#">project-level scopes</a>.</li> <li>• <a href="https://www.googleapis.com/auth/cloud-billing.readonly">https://www.googleapis.com/auth/cloud-billing.readonly</a></li> <li>• One of the following scopes: <ul style="list-style-type: none"> <li>– <a href="https://www.googleapis.com/auth/compute">https://www.googleapis.com/auth/compute</a></li> <li>– <a href="https://www.googleapis.com/auth/cloud-platform">https://www.googleapis.com/auth/cloud-platform</a></li> </ul> </li> </ul> | <p>The service account should have the Billing Account Viewer role and the following permissions to the project that stores billing data:</p> <ul style="list-style-type: none"> <li>• All <a href="#">project-level permissions</a></li> <li>• <code>billing.resourceAssociations.list</code></li> <li>• <code>billing.accounts.list</code></li> <li>• <code>compute.commitments.list</code></li> </ul> <p>For queries of billed costs and negotiated pricing via BigQuery, the service account should have the Billing Account Viewer role and the following permissions to the project that stores billing data:</p> <ul style="list-style-type: none"> <li>• <code>bigquery.tables.get</code></li> <li>• <code>bigquery.tables.getData</code></li> <li>• <code>bigquery.tables.list</code></li> <li>• <code>bigquery.jobs.create</code></li> </ul> |

### Action Execution Permissions

The following permissions are required to execute actions on GCP workloads.

| Minimum Authorization Scopes                                                                                                                                                                                                                                                                                                                                                                                   | Minimum IAM Roles/Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>• All <a href="#">project-level scopes</a></li> <li>• One of the following scopes: <ul style="list-style-type: none"> <li>– <a href="https://www.googleapis.com/auth/compute">https://www.googleapis.com/auth/compute</a></li> <li>– <a href="https://www.googleapis.com/auth/cloud-platform">https://www.googleapis.com/auth/cloud-platform</a></li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• All <a href="#">project-level permissions</a></li> <li>• <code>compute.instances.stop</code></li> <li>• <code>compute.instances.setMachineType</code></li> <li>• <code>compute.instances.start</code></li> <li>• <code>compute.disks.delete</code></li> <li>• <code>compute.zoneOperations.get</code></li> <li>• <code>compute.regionOperations.get</code></li> <li>• <code>compute.globalOperations.get</code></li> </ul> |

### Enabling Required GCP APIs

For Turbonomic to discover your GCP environment and billing details, you must enable the following APIs:

- Cloud Resource Manager API  
Creates, reads, and updates metadata for GCP resource containers.
- Compute Engine API  
Creates GCP VMs and volumes.
- Cloud Billing API  
Enables developers to manage billing for their GCP projects programmatically.
- BigQuery API  
A data platform for customers to create, manage, share, and query data.

To enable these APIs:

1. Navigate the GCP Console to the library of APIs.

On the GCP Console home page, navigate to **APIs & Services > Library**.

2. Search for the API you want to enable.

In the API Library **Search** box, enter the name of the API you want to enable. Then press **Enter** to execute the search. Repeat these steps for each of:

- Cloud Resource Manager API
- Compute Engine API
- Cloud Billing API
- BigQuery API

3. Enable the given API.

In the list that appears, click the API name to navigate to that API page. If the API is not already enabled, click **Enable**.

After you enable the given API, the console displays a details page for that API.

4. Navigate to the console Home page.

For each API you want to enable, navigate back to the home page and repeat these steps.

## GCP Billing

Turbonomic uses cost data in its analysis to make accurate recommendations for your workloads. To enable cost discovery and monitoring, you must grant your GCP service account "Billing Account Viewer" access to the related [billing accounts](#), and then add those billing accounts as targets.

When you add billing accounts, Turbonomic also discovers negotiated pricing and committed use discounts (CUD) for your workloads. The Discount Inventory chart in the user interface shows a list of discovered CUDs.

### Before Adding Targets

Review the following items before adding GCP Billing targets.

- **Service Accounts**

For instructions on creating service accounts in a `gcloud` shell session, see [GCP Target Service Account](#) and [GCP Billing Target Service Account](#).

- **Permissions**

To configure GCP targets, you create service accounts with roles that grant Turbonomic the permissions it needs to discover your GCP resources and costs, and to execute actions (optional).

For details, see [GCP Permissions](#).

### Adding Billing Accounts as Targets

To add a billing account target, specify the following:

- **Display Name**

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in GCP.

- **Service Account Key**

The [service account key](#) for the account associated with the billing account.

- **GCP Project ID**

The unique ID assigned to the project associated with the billing account. Costs accrued to this project are charged to the billing account you are adding.

- **BigQuery Settings**



BigQuery is a data warehouse that helps you manage GCP data. Turbonomic uses BigQuery resources to discover cost data for your environment. If you do not configure any of these fields, this target will not discover any cost data for Turbonomic analysis.

For more information, see:

- [BigQuery Resources](#)
- BigQuery Tables
  - [Example Queries for Cloud Billing Data Export](#)
  - [Schema - Standard Usage Cost Data](#)
  - [Schema - Pricing Data](#)

To configure your target to discover BigQuery data, specify values for the following fields.

- BigQuery Cost Export Data Set Name

The data set for billed costs. After you specify a data set, you must also specify the corresponding BigQuery Cost Export Table Name.

You can find the data set name in the GCP Billing dashboard under **Billing export / BIGQUERY EXPORT**.

- BigQuery Cost Export Table Name

This is the table of exported cost data. You can find the table name in the GCP BigQuery Explorer. Expand your project, and then expand the Cost Export Data Set Name.

- Enable Resource Level Detail From Cost Export Table

When you configure Billing Export, you can enable **Detailed usage cost**. To expose this detailed data to Turbonomic, turn on this option and then give the name of the detailed data table in the **BigQuery Cost Export Table Name** field. You can find the table name in the GCP BigQuery Explorer.

**Note:**

Only turn on this option if you have enabled **Detailed usage cost**. If you want to provide a **Standard usage cost** table, do not turn on this option.

- BigQuery Pricing Export Table Name

This field automatically populates with the table name used in BigQuery, `cloud_pricing_export`. You need to provide a different name if you do any of the following:

- Use a different table for negotiated pricing
- Specify a value for the **BigQuery Pricing Export Data Set Name** field. In that case, you must also specify the corresponding pricing export table.

- BigQuery Pricing Export Data Set Name

The data set for pricing. You can find the data set name in the GCP Billing dashboard under **Billing export / BIGQUERY EXPORT**.

- Billing Account ID

The identifier of the Billing Account you want to target. This field is required if you configure **BigQuery Pricing Export Data Set Name** and **BigQuery Pricing Export Table Name**.

- Proxy Information

Specify proxy information only if you connect to a GCP target via proxy.

- Proxy Host

The address of the proxy used for this target. Only fill out proxy information if you connect to the AppDynamics instance via a proxy.

- Proxy Port

The port to use with the proxy specified above. By default, this is 8080.

- Proxy Username  
The username to use with the proxy specified above.
- Proxy Password  
The password to use with the proxy specified above.
- Secure Proxy Connection  
When checked, Turbonomic will connect to the proxy via HTTPS.

## Actions

Turbonomic does not recommend actions for GCP Billing targets. However, the billing information will be used in conjunction with the GCP target to generate actions.

## Monitored Resources

Turbonomic does not monitor resources for GCP Billing targets.

## Microsoft Azure

Microsoft Azure is Microsoft's infrastructure platform for the public cloud. You gain access to this infrastructure through a service principal target. To specify an Azure target, you provide the credentials for the subscription and Turbonomic discovers the resources available to you through that service principal.

Through Azure service principal targets, Turbonomic automatically discovers the subscriptions to which the service principal has been granted access in the Azure portal. This in turn creates a derived target for each subscription that inherits the authorization provided by the service principal (e.g. contributor). You cannot directly modify a derived target, but Turbonomic validates the target and discovers its inventory as it does with any other target.

## Before Adding Targets

Review the following items before adding Azure service principal targets.

- **Azure Subscription Setup**

To configure an Azure service principal target, you must first set up your Azure subscription to grant Turbonomic the access it needs.

For details, see [Azure Subscription Setup](#).

- **Permissions**

For the service principal account, you must provide permissions to support all the actions you want to perform on any subscription accounts.

For details, see [Azure Permissions](#).

- **Azure Resource Manager**

Turbonomic requires Azure Resource Manager. This service provides the management layer that Turbonomic uses to discover and manage entities in your Azure environment.

**Note:**

Turbonomic does not discover Azure Classic VMs, as they do not utilize Azure Resource Manager.

- **Supported Regions**

Turbonomic supports discovery and management of entities in certain Azure regions, including Azure Government regions.

For details, see [Supported Azure Regions](#) and [Support for Azure Government](#).

- **Azure Reservations**

To manage the use of Azure reservations, the app registration for this target must have permissions to manage the reservations. In most cases, Reader permissions are sufficient.

For [scoped reservations](#), Turbonomic supports the *subscription* and *shared* scopes. To discover these reservations, you must assign the Azure App registration Reader permissions to both the Reservation and the Reservation Order (order ID). For details about reservation orders, see the Microsoft article, [Manage Reservations for Azure resources](#).

Turbonomic also discovers reservations scoped to a *resource group*, but treats them as shared (in the Discount Inventory chart, the scope for these reservations is shown as Shared\*). This could result in unreliable actions, such as scaling VMs within the resource group to other reservations, which could potentially increase costs. If you have existing reservations scoped to a resource group, be sure to change their scope in Azure before executing VM scale actions. For best results, change their scope to *shared*.

- **Collection of Memory Metrics**

We highly recommend enabling collection of memory metrics in your environment. Enabling memory metrics allows Turbonomic to generate Scale actions to optimize VM memory usage. For Turbonomic to collect memory metrics, you must enable the collection of these metrics on the VMs in your environment.

For details, see [Azure Memory Metrics Collection](#).

- **Support for Azure App Service**

Turbonomic discovers Azure App Service plans and app instances through your Azure targets.

For details, see [Support for Azure App Service](#).

- **Firewall and Proxy Access**

If you run Turbonomic behind a proxy or firewall, then you must configure Turbonomic to allow unrestricted access to certain URLs. This enables the full set of features for Azure Service Principal targets.

For details, see [Firewall and Proxy Access for Azure](#).

## **Adding Azure Service Principal Targets**

Be sure you have the following information at hand:

- Directory (tenant) ID and Application (client) ID

You should have noted the Directory (tenant) ID and Application (client) ID when you registered Turbonomic with Azure Active Directory.

For details, see [Registering Turbonomic with Active Directory](#).

- Client Secret

You should have saved the client secret generated for Turbonomic.

For details, see [Creating the Client Secret](#).

To add targets, select **Azure Service Principal** in the Target Configuration page, and provide the following information:

- Display Name

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in Azure.

- US Government

Turn this on to configure the target as an [Azure Government](#) target.

- Directory (Tenant) ID

The ID of the tenant that contains subscriptions to be managed with Turbonomic.

- Application (Client) ID  
The Client/App ID of the app registration that gives Turbonomic access to resources in your Azure subscription.
- Client Secret Key  
The client secret key for the app registration.
- Proxy Host  
The IP of your Proxy Host
- Proxy Port  
The port required for the proxy above
- Proxy User  
The username required for the proxy above
- Proxy Password  
The password required for the proxy above.
- Secure Proxy Connection  
When checked, Turbonomic will connect to the proxy via HTTPS.

**Note:**

After adding a target, allow at least 30 minutes for Turbonomic to fully discover resource groups in your environment.

**Adding a Billing Target**

In addition to service principal targets, you must also specify a billing target so Turbonomic can use custom pricing and discover reservations. When you configure a billing target, Turbonomic uses that richer pricing information to calculate workload size and reservation coverage for your Azure environment.

Currently, two billing targets are supported.

- Azure Billing – use this target if Turbonomic will manage *non-government* subscriptions.  
The Turbonomic Azure Billing target discovers your Azure billing account and *non-government* subscriptions. The target discovers the billing account and related subscriptions associated with your MCA Billing Account ID or EA Enrollment Number.  
For details, see [Azure Billing](#).
- Microsoft Enterprise Agreement – use this target if Turbonomic will manage *government* subscriptions.  
You can configure Turbonomic to manage *Azure government* subscriptions within the context of a Microsoft Enterprise Agreement (EA).  
For details, see [Microsoft Enterprise Agreement](#).

**Entity Mapping**

After validating your Azure targets, Turbonomic updates the supply chain with the entities that it discovered. The following table describes the entity mapping between Azure and Turbonomic.

| Azure                      | Turbonomic           |
|----------------------------|----------------------|
| Virtual Machine (VM)       | Virtual Machine (VM) |
| App Service - plan         | Virtual Machine Spec |
| App Service - app instance | App Component Spec   |

| Azure                                            | Turbonomic |
|--------------------------------------------------|------------|
| SQL Database (vCore or DTU)                      | Database   |
| Dedicated SQL Pool (for Azure Synapse Analytics) | Database   |
| Disk                                             | Volume     |
| Zone                                             | Zone       |
| Region                                           | Region     |

## Discovery of Entities in Locked Storage and Resource Groups

An Azure subscription can use locked storage or locked resource groups. For such subscriptions, Turbonomic discovers incomplete data. Locked resources affect Turbonomic discovery in either of these scenarios:

- Locked resource group

Turbonomic discovers all the entities in the resource group, but does not discover the resource group itself. For example, in the Top Accounts chart, the Resource Groups field will show no resource groups for a subscription that has a locked resource group.

- Locked storage

Turbonomic discovers all the entities in the resource group except the locked storage. It also discovers the resource group.

## Monitored Resources

### Note:

When you first configure an Azure target, under some circumstances the target has No Quotas Available, and so Turbonomic cannot discover the available templates. This can happen when you initially set up the Azure account and you have not enabled any providers. If this occurs, you can install a single VM in your cloud subscription to make quotas available. You can also navigate to the Azure Subscriptions Blade and select the subscription you want. For the resource providers, register the Microsoft.Compute option. For more information, see the following Microsoft article: [Resolve errors for resource provider registration](#).

Turbonomic monitors the following resources:

- **Virtual Machine**

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- I/O Throughput

I/O Throughput is the measurement of an entity's throughput to the underlying storage.

- **Virtual Machine Spec (App Service Plan)**

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Number of Replicas

Number of Replicas is the total number of VM instances underlying an App Service plan.

- **App Component Spec (App Service Instance)**

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- **Database**

The resources that Turbonomic can monitor depend on the pricing model in place for the given database entity.

- SQL Database - DTU Pricing Model

- DTU

DTU is the measurement of compute capacity for the database. DTU represents CPU, memory, and IOPS/IO Throughput bundled as a single commodity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- SQL Database - vCore Pricing Model

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- I/O Throughput

I/O Throughput is the measurement of an entity's throughput to the underlying storage.

- Dedicated SQL Pool (for Azure Synapse Analytics)

- DWU

DWU (Data Warehousing Unit) is the measurement of compute capacity for the dedicated SQL pool. DWU represents CPU, memory, and IO Throughput bundled as a single commodity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Connection

Connection is the measurement of database connections utilized by applications.

- **Volume**

- Storage Amount

Storage Amount is the storage capacity (disk size) of a volume.

Turbonomic discovers Storage Amount, but does not monitor utilization.

For a Kubeturbo (container) deployment that includes volumes, Kubeturbo monitors Storage Amount utilization for the volumes. You can view utilization information in the Capacity and Usage chart.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the measurement of IOPS capacity that is in use.

- I/O Throughput

I/O Throughput is the measurement of I/O throughput capacity that is in use.

## Actions

Turbonomic can recommend the following actions:

### Note:

Use the Potential Savings and Necessary Investments charts to view pending actions and evaluate their impact on your cloud expenditure.

- **Virtual Machine**

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing Azure reservations coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

If you need more capacity, then Turbonomic will recommend actions to purchase additional reservations.

- **Stop and Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

For details, see [Actions for Azure VMs](#).

- **Virtual Machine Spec (App Service Plan)**

- **Scale**

Scale Azure App Service plans to optimize app performance or reduce costs, while complying with business policies.

- **Delete**

Delete empty Azure App Service plans as a cost-saving measure. A plan is considered empty if it is not hosting any running apps.

For details, see [Virtual Machine Spec Actions](#).

- **App Component Spec (App Service Instance)**

### None

Turbonomic does not recommend actions for App Component Specs, but it does recommend actions for the underlying Virtual Machine Specs. For details, see [Virtual Machine Spec Actions](#).

- **Database**

### **Scale (SQL Database)**

- DTU Pricing Model

Scale DTU and storage resources to optimize performance and costs.

- vCore Pricing Model

Scale vCPU, vMem, IOPS, throughput and storage resources to optimize performance and costs.

### **Suspend (Dedicated SQL Pool)**

Suspend an idle dedicated SQL pool (used for Azure Synapse Analytics) to reduce compute costs.

#### **Note:**

Currently, Turbonomic does not generate actions to start a suspended pool. You can manually start a pool from Azure.

For details, see [Cloud Database Actions](#).

- **Volume**

- **Scale**

Scale attached volumes to optimize performance and costs.

- **Delete**

Delete unattached volumes as a cost-saving measure. Turbonomic generates an action immediately after discovering an unattached volume.

For details, see [Cloud Volume Actions](#).

## **Azure Subscription Setup**

To configure an Azure service principal target, you must first set up your Azure subscription to grant Turbonomic the access it needs.

Subscription must be registered with the Microsoft.Capacity resource provider. This can be done in the settings section of the Azure portal.

## **Azure Required Permissions and Roles**

Turbonomic interacts with Azure targets through an Azure Active Directory application/Service Principal. As part of subscription setup, you must assign a permission level through one of the built-in Azure roles.

For details about permissions and roles, see [Minimum Permissions and Roles for Azure](#).

## **Registering Turbonomic with Azure Active Directory**

The administrator of an Azure Active Directory (Tenant) can register an application with the tenant. This app registration gives an external application access to the tenant's resources. Turbonomic connects to an Azure target via an app registration.

For the latest information, refer to this [Azure article](#).

To register the application:

1. Log into the Azure Management Portal.
2. Add an App registration to an available tenant. The tenant ID will correspond with the tenant ID that you set for the target.
3. From the newly-created App registration blade in the **Management Portal**, go to the **Overview** blade. This blade displays the generated Directory (tenant) ID and Application (client) ID for this app.
4. From the newly created App registration blade in the **Management Portal**, go to the **Certificates and Secrets** blade.



This blade displays previously created certificates and client secrets, as well as provides the ability to create them.

## Creating the Client Secret

To create the client secret:

1. Navigate to the **Certificates and Secrets** section of your registered app.
2. Create a new client secret.
  - a. Click **+ New client secret**.
  - b. In the **Add a client secret** area, provide the required information.
    - Description – Provide a valid name.
    - Expires – Choose **Never**.
  - c. Click **Add**.

The blade generates a client secret for you to use when configuring the target in Turbonomic.

3. Copy the client secret.

### Important:

The client secret only displays *once*, during creation. It will no longer display after you leave the page.

## Enabling Turbonomic Access to Subscriptions

The final step in setting up your Azure subscription is to add the Active Directory Application (that you created above) to each and every subscription that you want Turbonomic to manage. Note that if you need to, you can assign different permissions to each subscription (for example, service principal).

Perform these steps for each subscription that Turbonomic will manage.

1. Navigate to your Azure subscriptions list.
2. Select the first subscription you want Turbonomic to manage.
3. Navigate to the Access control for that subscription.
4. Add a role. For details about roles, see [Minimum Permissions and Roles for Azure](#).
5. Add the application you registered with the tenant.

## Azure Permissions

### Minimum Permissions and Roles for Azure

You can use generic Azure roles to set up access for the account Turbonomic uses to access your target. This is an easy way to configure a target account, but you do not have full control over the access you grant to that account. For finer control, you should set the minimum permissions.

- Read-Only Access

Use the combined `Reader plus Storage Account Contributor` for the minimum combination required for Turbonomic to discover and access metrics across your Azure environment. `Storage Account Contributor` is required to access the Storage Account keys and establish a connection that can retrieve VM memory statistics.

- Action Execution Access

For action execution access, the user must have the `Owner` or `Contributor` role. `Contributor` is the least privileged role that enables Turbonomic to take actions on your Azure environment, including manually or automatically scaling VMs across instance types.

## Azure Service Principal and Subscription Permissions

For the service principal account, you must provide permissions to support all the actions you want to perform on any subscription accounts.

| Turbonomic Functionality                          | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Role validation                                   | <ul style="list-style-type: none"> <li>• <code>Microsoft.Authorization/roleAssignments/read</code><br/>Validates the role assigned to the Service Principal by checking if it has the minimum required permissions</li> <li>• <code>Microsoft.Authorization/roleDefinitions/read</code><br/>Queries the permissions list from the assigned custom role</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Discovery of subscriptions                        | <ul style="list-style-type: none"> <li>• <code>Microsoft.Resources/subscriptions/read</code><br/>Gets a list of accessible subscriptions for a tenant</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Discovery of resource groups, locations, and SKUs | <ul style="list-style-type: none"> <li>• <code>Microsoft.Resources/subscriptions/locations/read</code><br/>List all locations available for the subscriptions</li> <li>• <code>Microsoft.Resources/subscriptions/resourceGroups/read</code><br/>Discovers all resource groups for the subscriptions</li> <li>• <code>Microsoft.Compute/skus/read</code><br/>Gets a list of <code>Microsoft.Compute</code> SKUs available for your subscription</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Discovery of storage accounts                     | <ul style="list-style-type: none"> <li>• <code>Microsoft.Storage/storageAccounts/read</code><br/>Gets a list of storage accounts or gets the properties of a specific storage account</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Discovery of metrics for various entities         | <ul style="list-style-type: none"> <li>• <code>Microsoft.Insights/Metrics/Read</code><br/>Reads metrics for various resources from Azure Monitor</li> <li>• <code>Microsoft.OperationalInsights/workspaces/read</code><br/>Queries a list of Log Analytics workspaces. Certain metrics (such as VM memory) are fetched from these workspaces.</li> <li>• <code>Microsoft.OperationalInsights/workspaces/query/read</code><br/>Allows queries to data (such as metrics) stored in Log Analytics workspaces</li> <li>• <code>Microsoft.OperationalInsights/workspaces/query/InsightsMetrics/read</code><br/>Queries the <code>InsightsMetrics</code> Log Analytics table for VM memory metrics. Azure Monitor Agent must be configured to send memory metrics to the table.</li> <li>• <code>Microsoft.OperationalInsights/workspaces/query/Perf/read</code><br/>(Only required if the <code>Perf</code> Log Analytics table is configured for VM memory metrics, instead of the <code>InsightsMetrics</code> Log Analytics table)</li> </ul> |

| Turbonomic Functionality                         | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                  | <p>Queries the Perf Log Analytics table for VM memory metrics</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Discovery of VMs                                 | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachines/instanceView/read</code><br/>Gets the detailed runtime status of a VM and its resources</li> <li>• <code>Microsoft.Compute/virtualMachines/read</code><br/>Gets the properties of a VM</li> <li>• <code>Microsoft.Compute/virtualMachines/extensions/read</code><br/>(Only required if using storage account for VM memory configuration)<br/>Gets the properties of a VM extension, to detect diagnostics extension before fetching memory metrics from storage account</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Discovery of VM scale sets and availability sets | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/read</code><br/>Gets the properties of a VM scale set</li> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/read</code><br/>Retrieves the instance view of a VM in a scale set</li> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read</code><br/>Retrieves the properties of a VM in a scale set</li> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/virtualMachines/extensions/read</code><br/>(Only required if VM memory is configured onto a storage account table via a diagnostics agent extension)<br/>Gets the properties of an extension for a VM in a scale set</li> <li>• <code>Microsoft.Compute/availabilitySets/read</code><br/>Lists all availability sets and gets their properties</li> <li>• <code>Microsoft.Compute/availabilitySets/vmSizes/read</code><br/>Lists available sizes for creating or updating a VM in an availability set</li> </ul> |
| Discovery of reservations                        | <ul style="list-style-type: none"> <li>• <code>Microsoft.Capacity/reservationorders/reservations/read</code><br/>Monitors reservations data and reads all reservations</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Discovery of volumes                             | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/disks/read</code><br/>Gets the properties of volumes</li> <li>• <code>Microsoft.Storage/storageAccounts/listkeys/action</code><br/>(Only required if there are unmanaged disks in your environment)<br/>Discovers or queries unmanaged disks (volumes) in the storage account, and checks if unmanaged disks that became unattached are being deleted via a Turbonomic action</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| Turbonomic Functionality                                    | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Discovery of SQL databases (vCore/ DTU) and metrics         | <ul style="list-style-type: none"> <li>• <code>Microsoft.Sql/servers/read</code><br/>Lists all SQL servers in this subscription and gets their details</li> <li>• <code>Microsoft.Sql/servers/databases/read</code><br/>Lists and gets details about all SQL databases for all SQL servers in the subscription</li> <li>• <code>Microsoft.Sql/servers/databases/metrics/read</code><br/>Queries metrics for SQL databases</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Discovery of resources used in Azure Synapse Analytics      | <ul style="list-style-type: none"> <li>• <code>Microsoft.Synapse/SKUs/read</code><br/>Reads SKU details for a Synapse Analytics Service resource, such as SQL pools</li> <li>• <code>Microsoft.Synapse/workspaces/read</code><br/>Reads details about Synapse workspaces</li> <li>• <code>Microsoft.Synapse/workspaces/keys/read</code><br/>Gets details of Synapse workspace key definitions</li> <li>• <code>Microsoft.Synapse/workspaces/sqlDatabases/read</code><br/>Reads a list of Synapse SQL Analytics databases</li> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/read</code><br/>Reads a list of Synapse SQL Analytics pools</li> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/dataWarehouseUserActivities/read</code><br/>Reads user activities on Synapse SQL Analytics pools</li> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/extensions/read</code><br/>Gets extensions for Synapse SQL Analytics pools</li> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/operationStatuses/read</code><br/>Reads the results of asynchronous operations on Synapse SQL Analytics pools</li> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/usages/read</code><br/>Reads usage metrics for Synapse SQL Analytics pools</li> <li>• <code>Microsoft.Synapse/workspaces/sqlUsages/read</code><br/>Gets usage limits available for Synapse SQL Analytics pools</li> </ul> |
| Discovery of App Services (plans/app instances) and metrics | <ul style="list-style-type: none"> <li>• <code>Microsoft.Relay/namespaces/HybridConnections/read</code><br/>Lists all Service Bus Hybrid Connections used by web apps</li> <li>• <code>Microsoft.Web/geoRegions/Read</code><br/>Gets a list of available geographical regions for App Services</li> <li>• <code>Microsoft.Web/serverfarms/Read</code><br/>Lists and gets the properties of App Service plans</li> <li>• <code>Microsoft.Web/serverfarms/sites/read</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| Turbonomic Functionality                                        | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                 | <ul style="list-style-type: none"> <li>Gets a list of web apps that are part of App Service plans</li> <li>• <code>Microsoft.Web/serverfarms/skus/read</code></li> <li>Gets SKUs for App Service plans</li> <li>• <code>Microsoft.Web/sites/read</code></li> <li>Gets the properties of web apps that are part of App Service plans</li> <li>• <code>Microsoft.Web/sites/slots/Read</code></li> <li>Gets the properties of a web app deployment slot</li> <li>• <code>Microsoft.Web/serverfarms/metrics/read</code></li> <li>Queries metrics for App Services plans</li> <li>• <code>Microsoft.Web/serverfarms/usages/read</code></li> <li>Gets usage information for App Service plans</li> <li>• <code>Microsoft.Web/sites/metrics/read</code></li> <li>Gets metrics for web apps that are part of App Service plans</li> <li>• <code>Microsoft.Web/sites/usages/read</code></li> <li>Gets usage information for web apps that are part of App Service plans</li> </ul> |
| Discovery of clusters managed by Azure Kubernetes Service (AKS) | <ul style="list-style-type: none"> <li>• <code>Microsoft.ContainerService/managedClusters/read</code></li> <li>Discovers and lists managed container clusters</li> <li>• <code>Microsoft.ContainerService/managedClusters/agentPools/read</code></li> <li>Discovers agent pools on managed container clusters</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Discovery of desktop virtualization (VDI)                       | <ul style="list-style-type: none"> <li>• <code>Microsoft.DesktopVirtualization/hostpools/read</code></li> <li>Discovers Azure Desktop Virtualization host pools</li> <li>• <code>Microsoft.DesktopVirtualization/hostpools/sessionhosts/read</code></li> <li>Discovers session hosts in Azure Desktop Virtualization host pools</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Discovery of network resources                                  | <ul style="list-style-type: none"> <li>• <code>Microsoft.Network/networkInterfaces/read</code></li> <li>Lists and gets details about network interfaces for resources</li> <li>• <code>Microsoft.Network/publicIPAddresses/read</code></li> <li>Lists and gets details about public IP addresses for resources</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Discovery of cost data                                          | <ul style="list-style-type: none"> <li>• <code>Microsoft.Commerce/RateCard/read</code></li> <li>Discovers pricing information from the pay-as-you-go rate card; also returns offer data, resource/meter metadata, and rates for the given subscription</li> <li>• <code>Microsoft.Consumption/pricesheets/read</code></li> <li>Discovers pricing information from an Enterprise Agreement price sheet, and lists the price sheets data for a subscription or a management group</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

| Turbonomic Functionality                                     | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Discovery of locks that could prevent action execution       | <ul style="list-style-type: none"> <li>• <code>Microsoft.Authorization/locks/read</code><br/>Lists all locks for a subscription, and creates action prerequisites that may prevent action execution if locks prevent write operations</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Execution of actions for VMs                                 | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachines/deallocate/action</code><br/>Stops a VM to execute a disruptive action; powers off the VM and releases the allocated compute resources</li> <li>• <code>Microsoft.Compute/virtualMachines/powerOff/action</code><br/>Suspends a VM by powering it off. The VM will continue to be billed while suspended.</li> <li>• <code>Microsoft.Compute/virtualMachines/start/action</code><br/>Restarts a VM that was stopped to execute a disruptive action</li> <li>• <code>Microsoft.Compute/virtualMachines/vmSizes/read</code><br/>Lists available sizes that a VM can update to</li> <li>• <code>Microsoft.Compute/virtualMachines/write</code><br/>Updates a VM (for example, its size) as part of executing scale actions</li> <li>• <code>Microsoft.Network/networkInterfaces/join/action</code><br/>Allows a VM or VM scale set to rejoin its network after a scale action executes, by attaching the network interface to the VM</li> <li>• <code>Microsoft.KeyVault/vaults/deploy/action</code><br/>(Only required if a VM to be scaled is using Azure Key Vault)<br/>Enables access to secrets in a key vault when deploying Azure resources to the VM</li> <li>• Azure Compute Gallery images<br/>(Only required during VM scaling execution if VM images are located in a separate Azure Compute Gallery image, such as in a different subscription) <ul style="list-style-type: none"> <li>– <code>Microsoft.Compute/galleries/images/read</code><br/>Gets the properties of Azure Compute Gallery images</li> <li>– <code>Microsoft.Compute/galleries/images/versions/read</code><br/>Gets the versions for Azure Compute Gallery images</li> <li>– <code>Microsoft.Compute/galleries/read</code><br/>Gets Azure Compute Gallery images</li> </ul> </li> </ul> |
| Execution of actions for VM scale sets and availability sets | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/deallocate/action</code><br/>Stops a VM scale set to execute a disruptive action; powers off and releases the compute resources of the instances used by the VM scale set</li> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/start/action</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Turbonomic Functionality                               | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                        | <p>Restarts the VMs that were stopped to execute a disruptive action on VM scale set</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/vmSizes/read</code></li> </ul> <p>Lists available sizes for creating or updating a VM in a scale set</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/virtualMachineScaleSets/write</code></li> </ul> <p>Updates a VM scale set (for example, its size) as part of executing scale actions for VM scale sets</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Insights/AutoscaleSettings/Write</code></li> </ul> <p>Updates an autoscale setting as part of scale action execution</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Execution of actions for volumes                       | <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/disks/write</code></li> </ul> <p>Resizes or changes the storage tier of volumes</p> <p>Executes scale actions for volumes and re-attaches volumes to VMs after scaling</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Compute/disks/delete</code></li> </ul> <p>Deletes unattached volumes for managed disks</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete</code></li> </ul> <p>Deletes unmanaged disks that became unattached after the deletion of unattached volumes</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read</code></li> </ul> <p>Checks if a page blob exists before deleting unmanaged disks that became unattached, and returns the properties of an existing page blob</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Storage/storageAccounts/blobServices/containers/read</code></li> </ul> <p>Lists blob containers</p> <p>Finds containers hosting unmanaged disks that became unattached and are to be deleted</p> <ul style="list-style-type: none"> <li>• Recovery Services Vault</li> </ul> <p>(Only required if volumes are used for disaster recovery)</p> <p>Prevents the deletion of volumes used for disaster recovery, even if they become unattached</p> <ul style="list-style-type: none"> <li>– <code>Microsoft.RecoveryServices/Vaults/read</code></li> </ul> <p>Gets a list of Recovery Services Vaults containing replicated disks</p> <ul style="list-style-type: none"> <li>– <code>Microsoft.RecoveryServices/vaults/replicationProtectedItems/read</code></li> </ul> <p>Reads the Disk IDs of any protected items in the vaults</p> |
| Execution of actions for SQL databases (vCore and DTU) | <ul style="list-style-type: none"> <li>• <code>Microsoft.Sql/servers/databases/write</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

| Turbonomic Functionality                                                 | Required Permissions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                          | <p>Executes scale actions for DTU and vCore databases, and updates database properties</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Sql/servers/databases/pause/action</code><br/>Pauses a database as part of executing a scale or suspend action</li> <li>• <code>Microsoft.Sql/servers/databases/resume/action</code><br/>Resumes a paused database as part of executing a scale or suspend action</li> </ul>                                                                                                                                             |
| Execution of actions for dedicated SQL pools for Azure Synapse Analytics | <ul style="list-style-type: none"> <li>• <code>Microsoft.Synapse/workspaces/sqlPools/pause/action</code><br/>Pauses (suspends) Synapse SQL Analytics pools</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                        |
| Execution of actions for App Services (plans)                            | <ul style="list-style-type: none"> <li>• <code>Microsoft.Web/serverfarms/Delete</code><br/>Deletes an empty App Service plan (one that is not hosting any running apps)</li> <li>• <code>Microsoft.Web/serverfarms/Write</code><br/>Updates an App Service plan as part of a scale action</li> </ul>                                                                                                                                                                                                                                                                         |
| Uploads of plan results to Azure Migrate                                 | <p>These permissions are only required if you are <a href="#">uploading the results</a> of a 'Migrate to Cloud' plan to Azure Migrate.</p> <ul style="list-style-type: none"> <li>• <code>Microsoft.Migrate/migrateprojects/read</code><br/>Gets the properties of a Migrate project</li> <li>• <code>Microsoft.Migrate/migrateprojects/solutions/read</code><br/>Gets the properties of a Migrate project solution</li> <li>• <code>Microsoft.Migrate/migrateprojects/solutions/getconfig/action</code><br/>Gets the configuration of a Migrate project solution</li> </ul> |

For example, create a JSON file for your custom policy similar to the following, where **<Your\_subscription\_ID>** sets the role scope to the ID of the Azure subscription you want for this target:

```
{
 "Name": "Minimal Access",
 "Description": "Minimal access policy",
 "IsCustom": "true",
 "Actions": [
 "Microsoft.Authorization/locks/read",
 "Microsoft.Authorization/roleAssignments/read",
 "Microsoft.Authorization/roleDefinitions/read",
 "Microsoft.Capacity/reservationorders/reservations/read",
 "Microsoft.Commerce/RateCard/read",
 "Microsoft.Compute/availabilitySets/read",
 "Microsoft.Compute/availabilitySets/vmSizes/read",
 "Microsoft.Compute/disks/delete",
 "Microsoft.Compute/disks/read",
 "Microsoft.Compute/disks/write",
 "Microsoft.Compute/galleries/images/read",
 "Microsoft.Compute/galleries/images/versions/read",
 "Microsoft.Compute/galleries/read",
 "Microsoft.Compute/skus/read",
 "Microsoft.Compute/virtualMachineScaleSets/deallocate/action",
 "Microsoft.Compute/virtualMachineScaleSets/read",
 "Microsoft.Compute/virtualMachineScaleSets/start/action",
 "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/extensions/read",
 "Microsoft.Compute/virtualMachineScaleSets/virtualMachines/instanceView/read",
```



```

"Microsoft.Compute/virtualMachineScaleSets/virtualMachines/read",
"Microsoft.Compute/virtualMachineScaleSets/vmSizes/read",
"Microsoft.Compute/virtualMachineScaleSets/write",
"Microsoft.Compute/virtualMachines/deallocate/action",
"Microsoft.Compute/virtualMachines/extensions/read",
"Microsoft.Compute/virtualMachines/instanceView/read",
"Microsoft.Compute/virtualMachines/powerOff/action",
"Microsoft.Compute/virtualMachines/read",
"Microsoft.Compute/virtualMachines/start/action",
"Microsoft.Compute/virtualMachines/vmSizes/read",
"Microsoft.Compute/virtualMachines/write",
"Microsoft.Consumption/pricesheets/read",
"Microsoft.ContainerService/managedClusters/agentPools/read",
"Microsoft.ContainerService/managedClusters/read",
"Microsoft.DesktopVirtualization/hostpools/read",
"Microsoft.DesktopVirtualization/hostpools/sessionhosts/read",
"Microsoft.Insights/AutoscaleSettings/Write",
"Microsoft.Insights/Metrics/Read",
"Microsoft.KeyVault/vaults/deploy/action",
"Microsoft.Migrate/migrateprojects/read",
"Microsoft.Migrate/migrateprojects/solutions/getconfig/action",
"Microsoft.Migrate/migrateprojects/solutions/read",
"Microsoft.Network/networkInterfaces/join/action",
"Microsoft.Migrate/migrateprojects/read",
"Microsoft.Migrate/migrateprojects/solutions/getconfig/action",
"Microsoft.Migrate/migrateprojects/solutions/read",
"Microsoft.Network/networkInterfaces/join/action",
"Microsoft.Network/networkInterfaces/read",
"Microsoft.Network/publicIPAddresses/read",
"Microsoft.OperationalInsights/workspaces/query/InsightsMetrics/read",
"Microsoft.OperationalInsights/workspaces/query/Perf/read",
"Microsoft.OperationalInsights/workspaces/query/read",
"Microsoft.OperationalInsights/workspaces/read",
"Microsoft.RecoveryServices/Vaults/read",
"Microsoft.RecoveryServices/vaults/replicationProtectedItems/read",
"Microsoft.Relay/namespaces/HybridConnections/read",
"Microsoft.Resources/subscriptions/locations/read",
"Microsoft.Resources/subscriptions/read",
"Microsoft.Resources/subscriptions/resourceGroups/read",
"Microsoft.Sql/servers/databases/metrics/read",
"Microsoft.Sql/servers/databases/pause/action",
"Microsoft.Sql/servers/databases/read",
"Microsoft.Sql/servers/databases/resume/action",
"Microsoft.Sql/servers/databases/write",
"Microsoft.Sql/servers/read",
"Microsoft.Storage/storageAccounts/blobServices/containers/read",
"Microsoft.Storage/storageAccounts/listkeys/action",
"Microsoft.Storage/storageAccounts/read",
"Microsoft.Synapse/SKUs/read",
"Microsoft.Synapse/workspaces/keys/read",
"Microsoft.Synapse/workspaces/read",
"Microsoft.Synapse/workspaces/sqlDatabases/read",
"Microsoft.Synapse/workspaces/sqlPools/dataWarehouseUserActivities/read",
"Microsoft.Synapse/workspaces/sqlPools/extensions/read",
"Microsoft.Synapse/workspaces/sqlPools/operationStatuses/read",
"Microsoft.Synapse/workspaces/sqlPools/pause/action",
"Microsoft.Synapse/workspaces/sqlPools/read",
"Microsoft.Synapse/workspaces/sqlPools/usages/read",
"Microsoft.Synapse/workspaces/sqlUsages/read",
"Microsoft.Web/geoRegions/Read",
"Microsoft.Web/serverfarms/Delete",
"Microsoft.Web/serverfarms/Read",
"Microsoft.Web/serverfarms/Write",
"Microsoft.Web/serverfarms/metrics/read",
"Microsoft.Web/serverfarms/sites/read",
"Microsoft.Web/serverfarms/skus/read",
"Microsoft.Web/serverfarms/usages/read",
"Microsoft.Web/sites/metrics/read",
"Microsoft.Web/sites/read",
"Microsoft.Web/sites/slots/Read",
"Microsoft.Web/sites/usages/read"
],
"dataActions": [
 "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/delete",
 "Microsoft.Storage/storageAccounts/blobServices/containers/blobs/read"
],
"notActions": [],
"notDataActions": [],
"AssignableScopes": [
 "/subscriptions/<Your_subscription_ID>"
]

```

} ]

## Supported Azure Regions

Turbonomic supports discovery and management of entities in the following Azure regions:

| Region Code        | Region Name          | Notes |
|--------------------|----------------------|-------|
| eastus             | East US              |       |
| eastus2            | East US 2            |       |
| centralus          | Central US           |       |
| northcentralus     | North Central US     |       |
| southcentralus     | South Central US     |       |
| westcentralus      | West Central US      |       |
| westus             | West US              |       |
| westus2            | West US 2            |       |
| westus3            | West US 3            |       |
| canadaeast         | Canada East          |       |
| canadacentral      | Canada Central       |       |
| brazilsouth        | Brazil South         |       |
| northeurope        | North Europe         |       |
| westeurope         | West Europe          |       |
| francecentral      | France Central       |       |
| ukwest             | UK West              |       |
| uksouth            | UK South             |       |
| germanywestcentral | Germany West Central |       |
| norwayeast         | Norway East          |       |
| switzerlandnorth   | Switzerland North    |       |
| eastasia           | East Asia            |       |
| southeastasia      | Southeast Asia       |       |
| australiaeast      | Australia East       |       |
| australiasoutheast | Australia Southeast  |       |

| Region Code      | Region Name        | Notes                                |
|------------------|--------------------|--------------------------------------|
| australiacentral | Australia Central  |                                      |
| centralindia     | Central India      |                                      |
| southindia       | South India        |                                      |
| westindia        | West India         |                                      |
| japaneast        | Japan East         |                                      |
| japanwest        | Japan West         |                                      |
| koreacentral     | Korea Central      |                                      |
| koreasouth       | Korea South        |                                      |
| uaenorth         | UAE North          |                                      |
| southafricanorth | South Africa North |                                      |
| usgovarizona     | USGov Arizona      | Restricted Access (Azure Government) |
| usgovtexas       | USGov Texas        | Restricted Access (Azure Government) |
| usgovvirginia    | USGov Virginia     | Restricted Access (Azure Government) |

## Support for Azure Government

Azure Government provides dedicated regions for US government customers and their partners to architect secure cloud solutions and meet regulatory and compliance requirements.

Turbonomic discovers workloads in these regions when you add the required accounts as targets. Discovered workloads include VMs (including availability/scale sets), volumes, and SQL databases. Turbonomic recommends actions on these entities to address performance issues and optimize costs.

### Note:

Turbonomic currently does not support Azure Government integration with Application Insights. You can add accounts for Azure Government and Application Insights as targets, but Application Insights will only return performance data for non-government workloads.

## Prerequisites

When adding GovCloud targets, ensure the following prerequisites are met:

- The Azure Government account must be a service principal account. When you add the account, be sure to enable the **US Government** toggle to indicate that the account you are adding is for Azure Government.

For details about adding service principal accounts as targets, see [Microsoft Azure](#).

- In order for Turbonomic to discover custom pricing and reservations for Azure Government, you must add the corresponding Enterprise Agreement (EA) accounts.

For details about adding EA accounts as targets, see [Microsoft Enterprise Agreement](#).

### Note:

Currently, the [Azure Billing target](#) cannot be used to manage Azure Government accounts.

- To use Azure Government with a firewall or proxy, it must be configured to allow unrestricted access to certain URLs.

For details, see [Firewall and Proxy Access for Azure](#).

## Information in Charts

Use the following charts to view information about your government accounts and workloads.

- **Top Accounts** chart

Use the Top Accounts chart as a starting point.

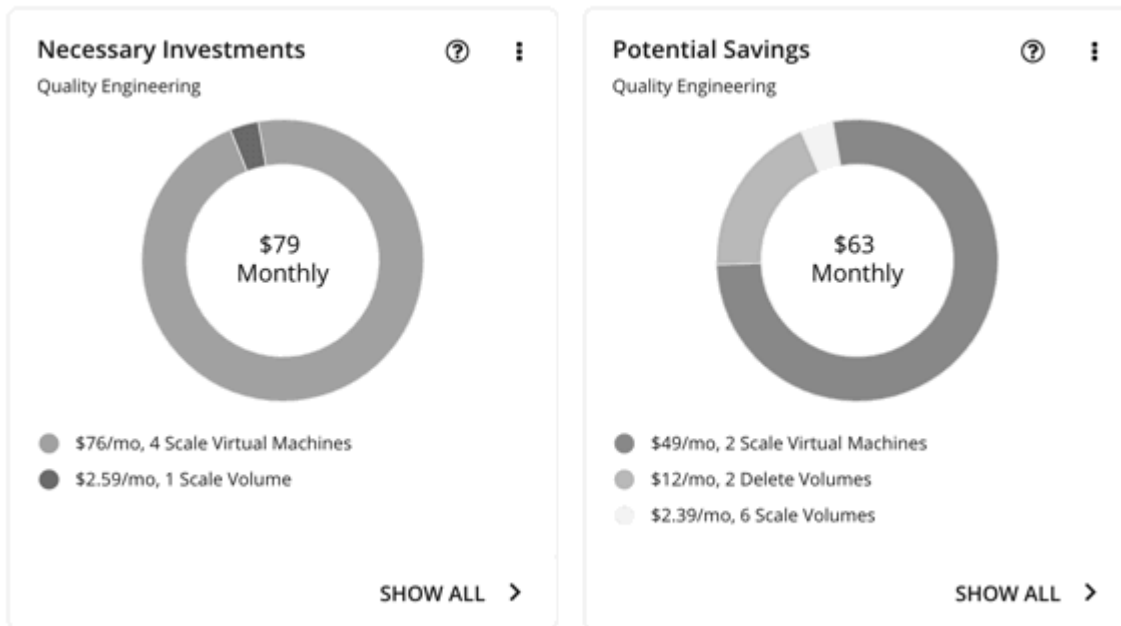
| Name                                                        | Worklo... | Potential Savings ⓘ | Actions    |
|-------------------------------------------------------------|-----------|---------------------|------------|
| Gov Pay-As-You-Go<br>Azure   US Government   ██████████     | 1         | \$0.96/mo           | 1 ACTION   |
| GovEA - Development 2<br>Azure   US Government   ██████████ | 1         | \$1.11/mo           | 1 ACTION   |
| ★ Development<br>AWS   GovCloud (US)   ██████████           | 9         | \$14/mo             | 4 ACTIONS  |
| Quality Engineering<br>AWS   GovCloud (US)   ██████████     | 16        | \$64/mo             | 17 ACTIONS |
| EA - ParkMyCloud<br>Azure   Global   ██████████             | 8         | \$132/mo            | 5 ACTIONS  |

SHOW ALL >

The chart shows the Azure Government subscriptions discovered via the service principal accounts that you have added as targets.

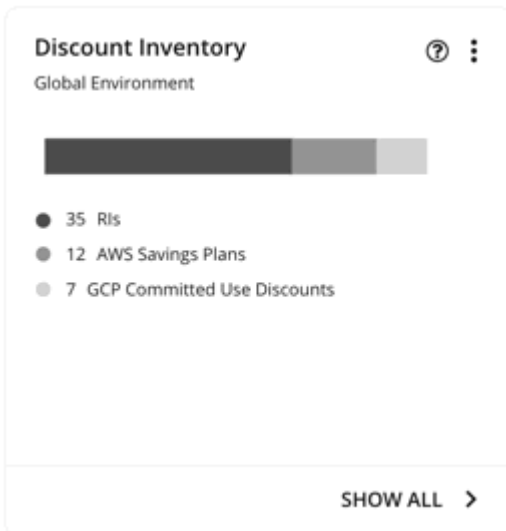
- **Necessary Investments** and **Potential Savings** charts

Set the scope to a government account or subscription, and then see the Necessary Investments and Potential Savings charts to evaluate the costs you would incur or save if you execute all the pending actions for your government workloads.



• **Discount Inventory** chart

The government accounts that you added as targets enable Turbonomic to gain full insight into the discounts that you have purchased for your government workloads. Even as you selectively add secondary targets, Turbonomic remains aware of all discounts, and how they are utilized across the board. This increases the accuracy of the allocation and purchase recommendations that Turbonomic generates for your government workloads.



**Workload Planning**

You can run an Optimize Cloud plan to identify performance and efficiency opportunities for existing government workloads, or a Migrate to Cloud plan to migrate government VM groups to another cloud provider.

For on-prem clusters, you can run a Migrate to Cloud plan to see how you can safely migrate the VMs in these clusters to a government account/subscription and region.

## Azure Memory Metrics Collection

We highly recommend enabling collection of memory metrics in your environment. Enabling memory metrics allows Turbonomic to generate Scale actions to optimize VM memory usage. For Turbonomic to collect memory metrics, you must enable the collection of these metrics on the VMs in your environment.

Turbonomic fetches memory metrics for Azure VMs using the following mechanisms, in order:

1. **Log Analytics Workspace - InsightsMetrics table** - (preferred method, supported by Azure) This mechanism is used when the VM is configured with the Azure Monitor Agent (AMA). This agent collects Guest OS Memory metrics and sends it to the Log Analytics Workspace InsightsMetrics table. Turbonomic then fetches these Available Memory metrics by querying this table.

For more information on the Azure Monitor Agent, see the [Azure documentation](#).

2. **Log Analytics Workspace - Perf table** - This mechanism is used when the VM is configured with the Log Analytics Agents (OMS/MMA) and the VM is connected to a Log Analytics Workspace. The Log Analytics Agents send the Guest OS Memory metrics to the Log Analytics Workspace Perf table. Turbonomic then fetches these Available Memory metrics by querying this table.

The Log Analytics Agents (OMS/MMA) will reach end of support by August 2024. For information on the recommended migration to the new Azure Monitor Agent (AMA), see the [Azure documentation](#).

3. **Storage Account SDK Based Diagnostics Monitoring** - This legacy VM memory collection mechanism installs an IaaS Diagnostics Agent on the VM, which sends guest OS level memory metrics to a designated Storage Account table. During discovery process, Turbonomic collects VM memory from these tables. If this mechanism is being used, Turbonomic needs relevant StorageAccount read permissions so that the memory metrics can be read from these tables.

For information about enabling this memory mechanism in the Azure Portal, see [How to Enable Memory Metrics on Azure Manually](#).

4. **Metrics REST API** - If memory could not be collected for a VM using the previously-mentioned mechanisms, Turbonomic tries to fetch host OS based memory for the VM via REST API calls, which is the same mechanism used for collecting other VM metrics such as Percentage CPU usage. Turbonomic collects VM memory and tries to generate memory-based scaling actions where appropriate.

### Note:

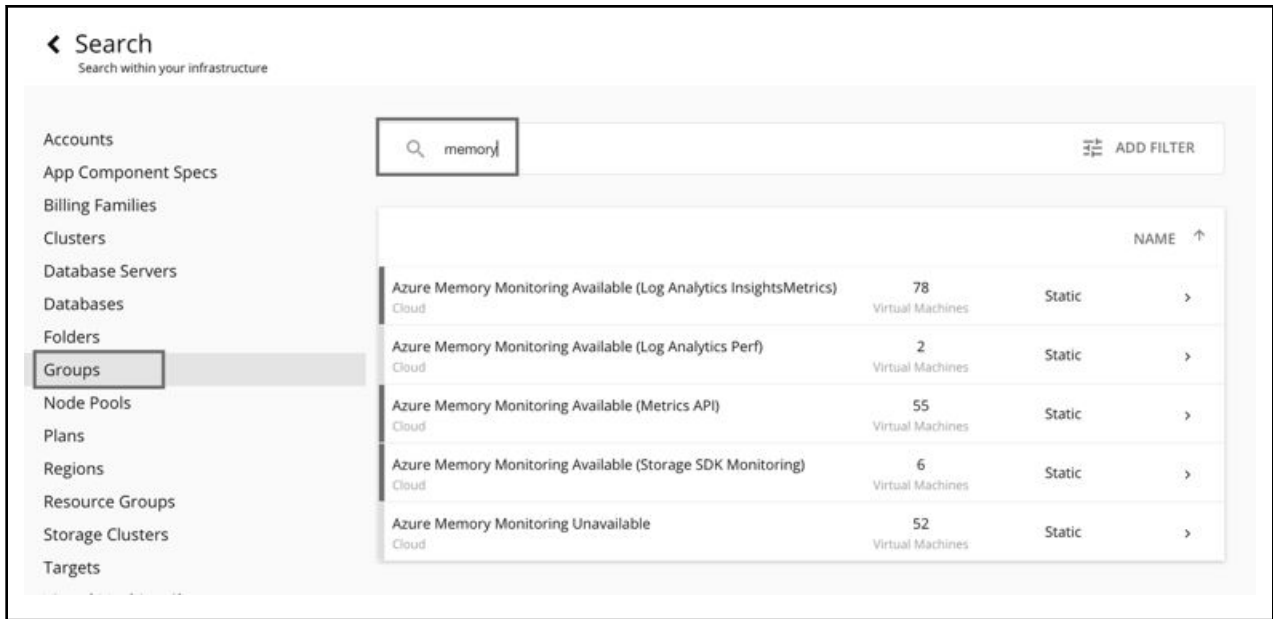
Azure provides Available Memory Bytes via REST API in Preview mode, or through the Azure Portal UI under the VM's **Monitoring > Metrics** section.

## Azure Memory Source Group Discovery

Azure Memory Source groups reflect the source from where VM memory is being fetched.

In the Turbonomic user interface, to search for a specific Azure Memory Source group, choose **Groups** in the Search page and enter memory in the search field. Each group contains the list of VMs for which memory was fetched via the specified mechanism. These are dynamic groups that get continuously updated based on each discovery.

The Azure Memory Monitoring Unavailable group contains VMs for which memory could not be fetched using any of the collection mechanisms. This group may contain VMs that are not in a RUNNING power state, and metrics data is often not available for these VMs. Memory-related scaling actions are not generated for VMs in this group; however, other non-memory Scale actions may be generated.



## Support for Azure App Service

Azure App Service is an HTTP-based service for hosting apps. With Azure App Service, app developers can easily create enterprise-ready apps and deploy them on a scalable and reliable cloud infrastructure.

Azure App Service offers several types of apps, including web apps, mobile apps, API apps, and logic apps. Each app runs as a set of *app instances* and is associated with a *plan* that defines compute resources (CPU, memory, and storage) available to the app.

When you add an Azure account as a target:

- Turbonomic discovers all the plans in that account, except App Service Environment v3 I4, I5, and I6. Plans appear as 'Virtual Machine Spec' entities in the supply chain.
- For plans associated with *web apps*, Turbonomic discovers the related app instances. In the supply chain, app instances appear as 'App Component Spec' entities. Turbonomic generates actions to scale these plans to optimize app performance.
- For plans associated with the other types of apps, Turbonomic does not generate scale actions or discover the related app instances.
- For plans that are not associated with any type of app, Turbonomic generates delete actions as a cost-saving measure.

For details about scale and delete actions, see [Virtual Machine Spec](#).

To discover plans and app instances, you must provide permissions to support all the actions you want to perform. For a list of permissions, see [Azure Permissions](#).

## Firewall and Proxy Access for Azure

If you run Turbonomic behind a proxy or firewall, then you must configure Turbonomic to allow unrestricted access to the following URLs. This enables the full set of features for Azure targets.

### Firewall and Proxy Access for Azure Service Principal

For Azure non-government:

- `ratecard.azure-api.net`
- `management.core.windows.net`
- `management.azure.com`
- `login.microsoftonline.com`

- ratecard.blob.core.windows.net
- api.loganalytics.io

For Azure Government:

- ratecard.azure-api.net
- management.core.usgovcloudapi.net
- management.usgovcloudapi.net
- login.microsoftonline.us
- ratecard.blob.core.usgovcloudapi.net
- api.loganalytics.us

## Firewall and Proxy Access for Azure Billing

- api.loganalytics.io

- login.microsoftonline.com

- management.azure.com

- For [Billed Costs](#):

- Cost Details API

```
*.blob.core.windows.net
```

- Cost export

- Less strict:

```
*.blob.core.windows.net
```

- Strict:

```
[STORAGE_ACCOUNT].blob.core.windows.net
```

Replace [STORAGE\_ACCOUNT] with the name of the storage account containing the cost export.

## Azure Billing

The Turbonomic Azure Billing target discovers your Azure billing account and *non-government* subscriptions. The target discovers the billing account and related subscriptions associated with your MCA Billing Account ID or EA Enrollment Number.

### Important:

The Microsoft Enterprise Agreement target has been deprecated and will eventually be removed from the product.

If you have previously added a Microsoft Enterprise Agreement target to manage *non-government* subscriptions, be sure to move to the Azure Billing target immediately. To do this, remove the existing target from the user interface, and then add the Azure Billing target. For details about deprecation, see [Microsoft Enterprise Agreement](#).

If you have Azure *government* subscriptions, use the Microsoft Enterprise Agreement target until further notice. Currently, the Azure Billing target cannot be used to manage government subscriptions. For details about adding a Microsoft Enterprise Agreement target, see [Microsoft Enterprise Agreement](#).

## Level of Support for MCA and EA Accounts

The following table describes the current level of support for MCA and EA accounts.



**Note:**

Empty Azure EA subscriptions that are not incurring any charges will not stitch with the Azure Billing target, and a discrepancy will occur in the offer ID of the subscription. Once the subscription incurs a charge, the stitching occurs and the subscription should correctly associate with the Azure Billing target with the correct offer ID.

| Item                                                     | MCA                                                                                                                                                                                                                                                                                                                                                                                                                                              | EA                                                                                                                                                                                             |
|----------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Required role to add the Azure Billing target            | Billing Account Reader role                                                                                                                                                                                                                                                                                                                                                                                                                      | Enrollment Reader role                                                                                                                                                                         |
| Discovery of pricing data                                | Pricing data is available for VMs, volumes, SQL databases (DTU and vCore), dedicated SQL pools (for Azure Synapse Analytics), and App Service plans (web apps, logic apps, and function apps).                                                                                                                                                                                                                                                   | Pricing data is available for VMs, volumes, SQL databases (DTU and vCore), dedicated SQL pools (for Azure Synapse Analytics), and App Service plans (web apps, logic apps, and function apps). |
| Discovery of Azure reservations                          | All reservations charged under the billing account are discovered. However, the percentage of workloads covered by reservations is currently not discovered.                                                                                                                                                                                                                                                                                     | All reservations charged under the billing account are discovered.                                                                                                                             |
| Usage of pricing data when recommending actions          | All actions for Azure workloads are supported, including actions to optimize/buy reservations.                                                                                                                                                                                                                                                                                                                                                   | All actions for Azure workloads are supported, including actions to optimize/buy reservations.                                                                                                 |
| Support for the Buy VM Reservations plan                 | The plan is fully supported.                                                                                                                                                                                                                                                                                                                                                                                                                     | The plan is fully supported.                                                                                                                                                                   |
| Scope for cost exports (if used to collect billed costs) | Cost exports must be created at the <i>Billing Profile</i> scope.<br><br>If you previously set up a cost export at the <i>Billing Account</i> scope, you must delete the cost export and then create a new one at the Billing Profile scope. Create this cost export for each active Billing Profile that falls under your MCA Billing Account. All cost exports you create for your active Billing Profiles are required to have the same name. | Cost exports must be created at the <i>Billing Account</i> scope.                                                                                                                              |
|                                                          | These scopes are <i>not</i> supported – Subscription, Management Group, and Resource Group.                                                                                                                                                                                                                                                                                                                                                      | These scopes are <i>not</i> supported – Subscription, Management Group, and Resource Group.                                                                                                    |

**Before Adding Targets**

Review the following items before adding Azure Billing targets.

- **Roles and Permissions**

To configure Azure Billing targets, you create service principal accounts with roles that grant Turbonomic the permissions it needs to discover your Azure billing resources and costs.

For EA accounts, the **Enrollment Reader** role is required.

- Only users with elevated access and the Enterprise Administrator role can apply the Enrollment Reader role to the service principal.
- The admin executing the put-assignment API must have elevated access in order to assign the Enrollment Reader at the tenant level.

For details, see the Microsoft documentation [here](#) and [here](#).

For MCA accounts, the **Billing Account Reader** role is required. To assign the role, navigate to **Cost Management + Billing > Access > Add role**. Select the role and assign it to the service principal.

- **Billed Cost Data**

Decide whether the target will collect billed cost data via the Azure [Cost Details API](#) or a cost export that you set up in your environment.

For details, see [Azure Billed Costs](#).

- **Firewall and Proxy Access**

If you run Turbonomic behind a proxy or firewall, then you must configure Turbonomic to allow unrestricted access to certain URLs. This enables the full set of features for Azure targets.

For details, see [Firewall and Proxy Access for Azure](#).

## Adding an Azure Billing Target

### Note:

Remove any corresponding non-government EA targets before adding an Azure Billing target.

For additional information about adding an Azure Billing target, see [Azure Billing Target Setup](#).

To add an Azure Billing target, select **Azure Billing** in the Target Configuration page, and provide the following information:

- Display Name

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in Azure.

- Billing Account ID

The ID of your billing account in Azure. You can find the billing ID in the **Cost Management + Billing** section of the Azure portal. The ID is different for MCA and EA accounts.

- MCA Billing Account ID

In the Azure portal, navigate to **Cost management + Billing** and then click your MCA billing account. The screen that displays includes a field for the Billing Account ID. For example, an ID could be:

095c24bb-6d32-5f7s-98e3-40123a4b56cd:2f98765e-a5b5

- EA Enrollment Number

In the Azure portal, navigate to **Cost management + Billing**, click to open the EA account, and navigate to **Properties**. The screen that displays includes a field for the Enrollment Number. For example, an Enrollment Number could be:

12345678

- Use a cost export

### Note:

Before turning this option on or off, read the guidelines in [Azure Billed Costs](#).

- Turned off: Turbonomic will collect data via the Cost Details API.

- Turned on: Turbonomic will collect data via a cost export.

If you turn on the setting, you must specify the name of the cost export.

To find the cost export name in the Azure portal, navigate to **Cost management + Billing** and then select **Exports**. A list of the cost exports you have created displays. For more information about setting up cost exports, see [Cost Export Setup](#).

- Directory (Tenant) ID

The ID of the tenant that contains subscriptions to be managed with Turbonomic. This should match the Directory ID that you give for the associated Service Principal target.

- Application (Client) ID

The Client/App ID found in App registration. This ID gives Turbonomic access to resources in your Azure subscription, and should match the Client/App ID that you specify for the associated Service Principal target.

- Client Secret Key

The secret key for the App Registration. This should match the secret key ID that you give for the associated Service Principal target.

- Proxy Host

The IP of your Proxy Host

- Proxy Port

The port required for the proxy above

- Proxy User

The username required for the proxy above

- Proxy Password

The password required for the proxy above.

- Secure Proxy Connection

When checked, Turbonomic will connect to the proxy via HTTPS.

**Note:**

It takes 24 hours after successfully adding a target for any Azure reservation information to display in the user interface.

## Actions

Turbonomic does not recommend actions for Azure Billing targets. However, the billing information will be used in conjunction with the Azure target to generate actions.

## Monitored Resources

Turbonomic does not monitor resources for Azure Billing targets.

## Azure Billed Costs

The [Azure Billing](#) target can collect billed cost data via the Azure [Cost Details API](#) or a cost export that you set up in your environment.

**Note:**

Support for the Cost Details API started in version 8.8.4. New deployments of the product will use this API by default.

For Azure Billing targets added in earlier releases of the product, any associated cost export will automatically carry over to version 8.8.4 or later. You can switch these targets to the Cost Details API any time.

| Consideration                                           | Cost Details API                                                                                                                                                                                                         | Cost Export                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Size of billed cost data                                | Microsoft recommends the Cost Details API for billed cost data that is less than 2 GB in size. Larger sizes will likely result in timeouts.<br><br>Turbonomic generates ad hoc cost exports when you choose this option. | Microsoft recommends a cost export for billed cost data that is at least 2 GB in size. The data export is in CSV format, and contains all the cost and usage data that Azure Cost Management collects.                                                                                                                                                                                                              |
| Prerequisites                                           | None                                                                                                                                                                                                                     | Set up a cost export. For details, see <a href="#">Cost Export Setup</a> .                                                                                                                                                                                                                                                                                                                                          |
| Firewall rule (if you run Turbonomic behind a firewall) | Allow unrestricted access to:<br><br><code>*.blob.core.windows.net</code><br><br>For a full list of firewall URLs, see <a href="#">Firewall and Proxy Access for Azure Billing</a> .                                     | Allow unrestricted access to one of the following URLs:<br><br>• Less strict:<br><br><code>*.blob.core.windows.net</code><br><br>• Strict:<br><br><code>[STORAGE_ACCOUNT].blob.core.windows.net</code><br><br>Replace [STORAGE_ACCOUNT] with the name of the storage account containing the cost export.<br><br>For a full list of firewall URLs, see <a href="#">Firewall and Proxy Access for Azure Billing</a> . |
| Target configuration in Turbonomic                      | Turn off the <b>Use a cost export</b> option.                                                                                                                                                                            | Turn on the <b>Use a cost export</b> option, and then specify the cost export name.                                                                                                                                                                                                                                                                                                                                 |

## Cost Export Setup

Skip this section if your Azure Billing target will collect billed cost data via the Cost Details API.

For billed cost data collected via a cost export, the Azure Billing uses Cost and Usage reports. In order for Turbonomic to display month-to-day expenses, you must set up a daily cost export of actual cost on the Azure portal with an export type of month-to-date costs.

The cost export must be created as follows:

- EA accounts – Create the cost export at the Billing Account scope.
- MCA accounts – Create the cost export at the Billing Profile scope.

### Note:

For MCA accounts, if you previously set up a cost export at the Billing Account scope, you must delete the cost export and then create a new one at the Billing Profile scope. Create this cost export for each active Billing Profile that falls under your MCA Billing Account. All cost exports you create for your active Billing Profiles are required to have the same name.

For both EA and MCA accounts, Subscription, Management Group, and Resource Group scopes are not supported.

In addition, you must provide the following permissions for the storage account and container associated with the cost export.

- Storage account – Reader and Data Access
- Container – Storage Blob Data Reader

We recommend creating a new cost export, even if you have an existing export that matches the setup noted here. Below is an example of a cost export setup.

- Export details
  - Metric  
Actual cost (Usage and Purchases)
  - Export type  
Daily export of month-to-date costs
- Storage
  - Use existing
  - Subscription  
EA-Development
  - Storage account  
turbocostexport
  - Container  
cost-export-container
  - Directory  
costExportDir

## Microsoft Enterprise Agreement

### Important:

The Microsoft Enterprise Agreement target has been deprecated and will eventually be removed from the product.

### Who is impacted?

Deprecation and removal impact customers who have previously added, or plan to add, a Microsoft Enterprise Agreement target to manage Azure *non-government* subscriptions.

- Customers who have previously added a Microsoft Enterprise Agreement target are advised to immediately remove the target from the user interface, and then add the Azure Billing target.
- Customers planning to add a Microsoft Enterprise Agreement target should use the Azure Billing target instead.

The Azure Billing target leverages the new cost management APIs and cost exports recommended by [Microsoft](#). For instructions on adding the Azure Billing target, see [Azure Billing](#).

Now that the Microsoft Enterprise Agreement target has been deprecated:

- No new features will be introduced.
- The following charts may not show historical cost data initially. Customers who notice any data loss in their environment should note that the charts will rebuild the data over time.
  - Workload Cost Breakdown
  - Cost Breakdown by Tag
  - Top Billed Cost by Account
  - Top Billed Cost by Service
  - Billed Cost by Service Provider

- The target will continue to be available for customers to add or use, and will continue to discover pricing data.
- Customers can continue to report problems with the target. Turbonomic Support will answer questions, help with configuration issues, and provide any known workarounds.

Details about the removal will be available later.

### **Who is *not* impacted?**

Deprecation and removal do *not* impact customers who have previously added, or plan to add, a Microsoft Enterprise Agreement target to manage Azure *government* subscriptions. Customers with these subscriptions should use the Microsoft Enterprise Agreement target until further notice. Currently, the Azure Billing target cannot be used to manage government subscriptions.

### **Overview of the Microsoft Enterprise Agreement Target**

You can configure Turbonomic to manage *Azure government* subscriptions within the context of a Microsoft Enterprise Agreement (EA).

An EA defines specific pricing, including the pricing for reservations. When you configure an EA target, and set the EA key to your Azure targets, Turbonomic uses that richer pricing information to calculate workload placement and reservations coverage for your Azure environment.

To enable Turbonomic management of Azure EA environments, you must configure:

- One Microsoft Enterprise Agreement target
- At least one Service Principal target that can discover the underlying Azure subscriptions. For information about Azure service principal targets, see [Microsoft Azure](#).

In the Targets View, you can identify the targets related to Azure EA as follows:



- EA Targets

The target that discovers the EA to track pricing and reservations. You can have one EA target per Turbonomic deployment.

- Azure Subscription Targets

The targets that manage the workloads in your Azure environment. These are discovered by Service Principal targets. Note that not all subscription targets *necessarily* participate in the EA. Expand these entries to see the related Service Principal target. For members of the EA, you can see the related EA target as well.

Subscriptions that do not participate in the EA appear as Standalone targets.

**Note:**

In rare circumstances, you can have a subscription that is not in use. The subscription has no workloads associated with it. In this case, Turbonomic identifies the subscription as Standalone. This is because the target cannot discover any cost or usage information that would relate the subscription to its EA.

Empty Azure EA subscriptions that are not incurring any charges will not stitch with the Azure Billing target or the Azure EA target, and a discrepancy will occur in the offer ID of the subscription. Once the subscription incurs a charge, the stitching occurs and the subscription should correctly associate with the Azure Billing target with the correct offer ID.

- Service Principal Targets

The Azure target that you configure to discover Azure subscription targets. Expand the entry to see the discovered targets. If you have configured an EA target, the entry lists that as well, along with the EA enrollment number.

## Reservations and Azure EA

For Azure environments, Turbonomic can only discover and use reservations if you have configured a Microsoft Enterprise Account target, and if one or more subscriptions participate in that EA.

To discover and manage reservations in Azure environments, Turbonomic uses both the EA target and the associated subscription targets. On its own, a subscription target exposes costs for pay-as-you-go pricing. The EA target discovers pricing for the available reservations. Turbonomic combines this information to track:

- Utilization of reservations
- VMs covered by reservations
- VM costs (accounting for reservations)
- Purchase recommendations

### Note:

This release of Turbonomic does not support discovery and management of reservations for Classic VMs, Classic Cloud Services, and Suppressed Core VMs.

## Cost Calculations for Azure Environments

To understand the reported costs in your Azure environment, consider these points:

- For targets that participate in the EA, Turbonomic uses the terms of the given EA, and bases costs on the Offer ID that is effective for the given subscription.
- For VMs in Azure, reservations pricing does not include the cost of the OS license. However pricing for on-demand VMs does include the license cost.

### Note:

For Microsoft Azure EA environments, the projected cost for actions to purchase reservations might not match associated costs you find in the Microsoft Pricing Calculator.

Turbonomic actions can recommend purchases. For these recommendations, the action assumes a free Linux OS, so the cost estimate does not include the OS cost. However, The Microsoft Pricing Calculator does include costs for OS licenses. As a result, when you compare the Turbonomic cost estimates to the values in the Pricing Calculator, it's likely that the two estimates will not match. This difference also affects the Break Even Point that appears in the Recommended RI Purchases chart. Because the recommended purchases do not include Azure costs for OS licenses, the listed Break Even Point can be optimistic.

- For on-prem workloads you migrated to Azure, Turbonomic recognizes Azure Hybrid Benefit (AHUB) savings for reservations and on-demand workloads. The costs you see in Turbonomic charts include this benefit. However, remember that recommended actions do not include any license cost, so the actions will not reflect any proposed AHUB savings (see above).

## Before Adding Targets

Review the following items before adding Azure Billing targets.

- **Access Key and Enrollment Number**

A Microsoft Azure EA access key and an EA enrollment number are required to add the target.

- **Firewall and Proxy Access**

Your firewall must allow access to the `consumption.azure.com:443` address.

- **Access to Cost Data**

Be sure to enable access to your cost data. You can enable access in the Azure portal or in the EA portal.

### Note:

For additional information, see [Assign access to Cost Management data](#).



## Enabling Access to Cost Data via the Azure Portal

1. Log into to the Azure portal (<https://portal.azure.com>).

You should log in with an enterprise administrator account.

2. Open your Billing Account.

Navigate to **Cost Management + Billing | Billing scopes**. Select your billing account from the list of available accounts.

3. Configure cost access.

In the **Settings** group, select **Policies**. Then turn on **DEPARTMENT ADMINS CAN VIEW CHARGES** and **ACCOUNT OWNERS CAN VIEW CHARGES**.

## Enabling Access to Cost Data via the EA Portal

1. Log into to the Azure portal (<https://ea.azure.com>).

You should log in with an enterprise administrator account.

2. Select **Manage** in the left-hand navigation pane

3. Configure cost access.

For the cost management scopes that you want to provide access to, enable the charge option to **DA view charges** and/or **AO view charges**.

### Note:

After you configure cost access, most scopes also require Azure role-based access control (Azure RBAC) permission configuration in the Azure portal.

If you just enabled these settings, it can take up to 24 hours for the changes to take effect. For more information, see [Troubleshoot enterprise cost views](#) in the Microsoft Azure documentation.

If you perform self-service exchanges for your reservations, Turbonomic does not discover the new charges for the exchanged reservations through the Azure EA target. To track the charges after you have exchanged reservations, ensure you have an Azure subscription target for the affected scope of Azure workloads, and that subscription has read access to reservations information.

## Adding Microsoft Enterprise Agreement Targets

To add targets, select **Microsoft Enterprise Agreement** in the Target Configuration page, and provide the following information:

- Target Name

The name that identifies the target in Turbonomic. This name is for display purposes only and does not need to match any name in Azure.

- Enrollment Number

The EA enrollment number (found in your EA admin account at [ea.azure.com](https://ea.azure.com))

- API Key

The API Access Key for the EA (found in your EA admin account at [ea.azure.com](https://ea.azure.com))

- Proxy Host

The IP address of the proxy server used, if any

- Proxy Port

The port number of the proxy server

- Proxy User

The username of the proxy user used to authenticate

- Proxy Password

The password of the proxy user used to authenticate

When you add the target and it validates, Turbonomic:

- Recognizes any existing Azure targets in your environment that are part of the EA
- Updates these targets with custom prices from the EA
- Discovers reservations in these targets

**Note:**

This can take up to 24 hours, as target stitching occurs after the next bill processing cycle.

## Actions

Turbonomic does not recommend actions for Microsoft Enterprise Agreement targets. However, the billing information will be used in conjunction with the Azure target to generate actions.

## Monitored Resources

Turbonomic does not monitor resources for Microsoft Enterprise Agreement targets.

## Storage Targets

---

Adding a storage Target enables Turbonomic to connect to your storage subsystem through a native or SMI-S provider API. Turbonomic uses the target's API to access and collect information from each of the underlying disk arrays. The information is used to set disk performance characteristics according to the type and capacity of storage, leading to improved workload placement.

Similarly, Turbonomic determines the relationships between storage controllers and disk arrays, and the location of datastores within those arrays. This information also helps optimize workload placement at a more granular level.

For on-premises applications, this optimization will enable Turbonomic to make more informed decisions about which storage devices the workloads hosting your applications run on, and assist in assuring application SLO. In the cloud, storage data is handled as part of the public cloud target.

Both virtual machines and containers benefit from this level of optimization. In the case of short-lived containers, Turbonomic will suggest the best datastore to hold persistent data, and paired with a container or hypervisor target, will select the optimal match of compute and storage resources. For longer-lived containers and virtual machines, each workload will be continually assessed for SLA/SLO, and recommendations to move or resize storages will ensure the continued efficiency of your environment.

The section below describes the storage supply chain. For information on how to add specific storage targets, the resources Turbonomic can monitor for the various supply chain entities, and the actions it can take to optimize the environment, refer to the target configuration instructions for your specific storage type.

## Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities in turn host Storage entities (datastores).

## Entity Mapping

| Turbonomic Mapping | EMC VMAX                     | EMC XtremIO | HPE 3Par       | NetApp | Nutanix   | Pure   |
|--------------------|------------------------------|-------------|----------------|--------|-----------|--------|
| Storage            | Volume (Regular, Thin, Meta) | Volume      | Virtual Volume | Volume | Container | Volume |

| <b>Turbonomic Mapping</b> | <b>EMC VMAX</b>         | <b>EMC XtremIO</b> | <b>HPE 3Par</b> | <b>NetApp</b>      | <b>Nutanix</b> | <b>Pure</b> |
|---------------------------|-------------------------|--------------------|-----------------|--------------------|----------------|-------------|
| Disk Array                | Disk Group or Thin Pool | XTremIO Cluster    | CPG             | Aggregate          | Storage Pool   | Shelf Array |
| Storage Controller        | VMAX Array              | XTremIO Cluster    | Controller      | Controller / Filer | Controller VM  | Controller  |

## **Actions**

Turbonomic recommends actions for storage targets as follows.

### **Note:**

This is a general list of actions for storage managed by storage controllers. The actions that Turbonomic can recommend or automate depend on the actual technology. For example, Turbonomic can automate a datastore move across disk arrays or storage controllers for NetApp in C mode, but not for other storage technologies.

You can see how actions differ per technology in each section that describes adding a specific type of Storage Manager target.

- Storage
  - Start Storage
  - Provision Storage
  - Suspend Storage
  - Move (only with Storage Targets configured)
  - Resize (only with Storage Targets configured)
- Disk Arrays
  - Provision Disk Array
  - Start Disk Array
  - Suspend Disk Array
  - Move Disk Array (for NetApp Cluster-Mode, only)
  - Move Virtual Machine
  - Move Datastore
- Storage Controller
  - Provision Storage Controller (recommendation only)

## **Monitored Resources**

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.

### **Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency
  - The utilization of latency on the datastore.
- Disk Array
  - Storage Amount
    - The utilization of the Disk Array's capacity.
  - Storage Provisioned
    - The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the disk array.
  - Latency
    - The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller

**Note:**

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- CPU
  - The utilization of the Storage Controller's allocated CPU.
- Storage Amount
  - The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## EMC VMAX

Turbonomic supports management of VMAX2 and 3 Series storage arrays. The VMAX series is a family of enterprise storage arrays designed for SAN environments. Turbonomic connects to VMAX storage systems via an EMC SMI-S provider that has the disk arrays added to it. A single SMI-S provider can communicate with one or more disk arrays. When you specify an SMI-S provider as a target, Turbonomic discovers all the added disk arrays.

**Note:**

Turbonomic does not utilize Unisphere. Data is collected exclusively from the SMI-S provider.

Turbonomic will create Storage Groups based on the SLO levels defined in VMAX3 Targets. By default, Storage vMotion actions will respect these SLO levels based on the configured response time.

### Prerequisites

- EMC SMI-S Provider V8.x
- A service account that Turbonomic can use to connect to the EMC SMI-S Provider (typically the default admin account)

### Adding VMAX Targets

To add VMAX targets, select the **Storage > VMAX** option on the Target Configuration page and provide the following information:

- Address

The IP or host name of the SMI-S provider. If the provider address begins with https, you must follow the IP with the port used to connect.

- Use Secure Connection

If checked, port 5989 will be used to connect. If unchecked, port 5988 will be used.

- Username

The Username for the SMI-S provider.

- Password

The Password for the SMI-S provider.

## Entity Comparison

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in EMC VMAX to those used in Turbonomic:

| EMC VMAX Name                                                    | Turbonomic Entity  |
|------------------------------------------------------------------|--------------------|
| Volume (Regular, Thin, Meta)                                     | Storage            |
| Storage Resource Pool (VMAX3) / Thick Provisioned Pool (earlier) | Disk Array         |
| Storage Group (VMAX3) / Thin Provisioned Pool (earlier)          | Logical Pool       |
| VMAX Array                                                       | Storage Controller |

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type  | Can Be Automated                  | Recommendations only    |
|--------------|-----------------------------------|-------------------------|
| Storage      | Provision (Cloning), Delete, Move | Resize (V-Volumes only) |
| Logical Pool |                                   | Resize                  |

## Monitored Resources

When calculating available storage, Turbonomic excludes disks devoted to the VMAX operating system by default. If these disks are assigned to new raid groups or storage pools, the capacity of those disks will then be considered when calculating the capacity of the Storage Controller.

Turbonomic monitors the following storage resources:

- Storage

- Storage Amount

The utilization of the datastore's capacity.

- Storage Provisioned

The utilization of the datastore's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency  
The utilization of latency on the datastore.
- Logical Pool
  - Storage Amount  
The utilization of the logical pool's capacity.
  - Storage Provisioned  
The utilization of the logical pool's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the logical pool.
  - Latency  
The utilization of latency on the logical pool.
- Disk Array
  - Storage Amount  
The utilization of the Disk Array's capacity.
  - Storage Provisioned  
The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the disk array.
  - Latency  
The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller
  - Storage Amount  
The utilization of the storage controller's capacity.

## EMC XtremIO

EMC® XtremIO® is a flash-based (SSD) storage solution, designed to push data to applications at higher speeds. The system building blocks are SAN appliances called X-Bricks. A deployment is organized into clusters of X-Bricks, and the clusters are managed by the XtremIO Management Server (XMS).

Turbonomic connects to X-Bricks through the XMS. The XMS presents a unified view of each connected X-Brick cluster, rather than exposing the individual X-Bricks within each cluster. Within Turbonomic, each X-Brick cluster displays as a single storage controller with an associated disk array.

The relationship between Storage entities and individual X-Bricks within the cluster is not exposed through the XMS — Turbonomic cannot make recommendations to move datastores from one X-Brick to another. Additionally, the X-Brick has a fixed form factor — Turbonomic does not recommend resize actions for disk array or storage controller resources.

Turbonomic recognizes XtremIO arrays as flash storage and sets the IOPS capacity on discovered arrays accordingly.

### Prerequisites

- A service user account on XMS 4.0 or higher — typically the default `xmsadmin` account

Turbonomic uses this account to connect to the XMS and execute commands through the XtremIO API.

## Adding XtremIO Targets

For EMC XtremIO targets, select the **Storage > EMC XtremIO** option on the Target Configuration page and provide the following information:

- Address  
The name or IP address of the XtremIO Management Server (XMS).
- Username/Password  
Credentials for a user account on the XMS.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in XtremIO to those used in Turbonomic:

| <b>XTremIO Name</b> | <b>Turbonomic Entity</b> |
|---------------------|--------------------------|
| Volume              | Storage                  |
| XTremIO Cluster     | Disk Array               |
| XTremIO Cluster     | Storage Controller       |

## Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory [Storage Supply Chain](#).

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| <b>Entity Type</b> | <b>Can Be Automated</b> | <b>Recommendations only</b> |
|--------------------|-------------------------|-----------------------------|
| Storage            |                         | Provision, Resize Up        |
| Disk Array         |                         |                             |
| Storage Controller |                         | Provision                   |

## Monitored Resources

When calculating available storage, Turbonomic excludes disks devoted to the VNX operating system.

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount  
The utilization of the datastore's capacity.
  - Storage Provisioned  
The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency
  - The utilization of latency on the datastore.
- Disk Array
  - Storage Amount
    - The utilization of the Disk Array's capacity.
  - Storage Provisioned
    - The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the disk array.
  - Latency
    - The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller

**Note:**

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- CPU
  - The utilization of the Storage Controller's allocated CPU.
- Storage Amount
  - The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## EMC ScaleIO

EMC ScaleIO is an example of Software-Defined Storage for the datacenter. It creates a Virtual SAN overlaying commodity infrastructure that consists of multiple LAN-connected Servers with locally attached commodity Storage. It presents a standard Block Storage interface to Applications accessing the Virtual SAN.

Turbonomic communicates with the EMC ScaleIO system via the REST API Gateway.

### Prerequisites

- EMC ScaleIO 2.x or 3.x
- A service account that Turbonomic can use to connect to the ScaleIO Gateway.

### Adding EMC ScaleIO Targets

To add EMC ScaleIO targets, select the **Storage > EMC ScaleIO** option on the Target Configuration page and provide the following information:

- Address
  - The IP or host name of the Gateway.
- Username
  - The Username for the Gateway service account.



- Password

The Password for the Gateway service account.

## Entity Comparison

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in EMC ScaleIO to those used in Turbonomic:

| EMC ScaleIO Name  | Turbonomic Entity  |
|-------------------|--------------------|
| Volume            | Storage            |
| Storage Pool      | Disk Array         |
| Protection Domain | Storage Controller |

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type       | Can Be Automated    | Recommendations only         |
|-------------------|---------------------|------------------------------|
| Storage           | Provision (Cloning) | Resize (Disabled by default) |
| Disk Array        |                     | Resize Disk Array            |
| Protection Domain |                     | Provision (Cloning)          |

## Monitored Resources

Turbonomic monitors the following storage resources:

- Storage

### Note:

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- Storage Amount

The utilization of the datastore's capacity.

- Storage Provisioned

The utilization of the datastore's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

- Disk Array

- Storage Amount

The utilization of the Disk Array's capacity.

- Storage Provisioned

The utilization of the Disk Array's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

- The summation of the read and write access operations per second on the disk array.
- Latency
  - The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller
  - Storage Amount
    - The utilization of the storage controller's capacity.

## EMC VPLEX

Turbonomic supports management of EMC VPLEX virtual storage systems in a local configuration, via the VPLEX API. Currently, Turbonomic does not support Metro or Geo configurations.

VPLEX is used to aggregate and refine data collected between connected Storage and Hypervisor targets. VPLEX supports one-to-one, one-to-many, and many-to-one relationships between virtual volumes and LUNs. Only one-to-one mapping between virtual volume and LUNs is supported by Turbonomic.

### Prerequisites

- VPLEX Management Server
- Hypervisor target supported by Turbonomic
- Storage target supported by Turbonomic

#### Note:

In order for Turbonomic to make use of the information provided by VPLEX, you must also add the hypervisor and storage layered under it as targets.

### VPLEX Permissions

| Turbonomic Functionality | Required Permissions                                                |
|--------------------------|---------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"> <li>• Service Account</li> </ul> |
| Action Execution         | <ul style="list-style-type: none"> <li>• Admin account</li> </ul>   |

### Adding EMC VPLEX Targets

To add EMC VPLEX targets, select the **Storage > EMC VPLEX** option on the Target Configuration page and provide the following information:

- Address:
  - The IP or Hostname of the VPLEX Management Server
- Username:
  - The Username for the VPLEX Management Server
- Password:
  - The Password for the VPLEX Management Server
- Port Number:
  - The port number for the remote management connection. The default port number for the VPLEX Management server is 443
- Secure Connection:
  - Select this option to use a secure connection (HTTPS)

#### Note:

The default port (443) uses a secure connection.

## Supported Actions

For this target, actions are generated and executed via the underlying storage targets. Turbonomic will use the enhanced visibility provided by VPLEX to make more intelligent storage decisions- for example, recommending storage vMotion between pools.

## HPE 3PAR

HPE 3PAR StoreServ systems use controller nodes to manage pools of storage resources and present a single storage system to consumers. Turbonomic communicates with the HPE 3PAR system via both the WSAPI and SMI-S providers that are installed on the 3PAR controller node.

### Prerequisites

- SMI-S Provider enabled and configured on the controller node.
- WSPAI Provider enabled and configured on the controller node.
- A service account on the controller node that Turbonomic can use to connect to the SMI-S and WSPAI providers.

#### Note:

For discovery and monitoring, the Turbonomic service account must have the Browse permission on all monitored domains. To exclude domains from monitoring, the Turbonomic service account must have no permissions on those domains. For action execution, Turbonomic requires the Edit permission.

### Setting Up the SMI-S Provider

The HPE 3PAR SMI-S Provider should be installed on the controller node. It is disabled by default — you must ensure that it is installed properly and running on the controller node.

To enable the SMI-S provider:

1. Log into the HPE 3PAR Command Line Interface (CLI).

Open a secure shell session (ssh) on the controller node. Default credentials are 3paradm/3pardata.

2. Check the current status of the SMI-S provider.

In the shell session, execute the command, `showcim`.

3. If the CIM service is not running, start it.

Execute the command `startcim` to enable the CIM service and the SMI-S provider.

To stop the SMI-S provider, execute the command `stopcim -f -x`.

### Setting Up the WSAPI Provider

The HPE 3PAR WSAPI Provider should be installed on the controller node.

To enable the WSAPI provider:

1. Log into the HPE 3PAR Command Line Interface (CLI).

Open a secure shell session (ssh) on the controller node. Default credentials are 3paradm/3pardata.

2. Check the current status of the WSAPI provider.

In the shell session, execute the command, `showwsapi`.

3. If the WSAPI service is not running, start it by executing the command `startwsapi`.

Execute the command `set wsapi -http enable` to allow only insecure connections, or `set wsapi -https enable` to allow only secure connections.

To stop the WSAPI provider, execute the command `stopwsapi -f`.

## Adding HPE 3PAR Targets

To add an HPE 3PAR target, select the **Storage > HPE 3Par** option on the Target Configuration page and provide the following information:

- Address

The name or IP address of the 3PAR controller node.

By default, the controller provides SMI-S data over port 5988 (HTTP) or port 5989 (HTTPS). If your installation uses a different port for SMI-S, include the port number in the Address field.

- Username/Password

Credentials for a user account on the controller node.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in HPE 3PAR to those used in Turbonomic:

| HPE 3PAR Name    | Turbonomic Entity  |
|------------------|--------------------|
| Virtual Volume   | Storage            |
| CPG              | Disk Array         |
| AO Configuration | Logical Pool       |
| Controller       | Storage Controller |

## Supply Chain

Storage targets (storage controllers) add Storage Controller, Logical Pool and Disk Array entities to the supply chain. Logical Pool and Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory [Storage Supply Chain](#).

## 3Par Adaptive Optimization

Adaptive Optimization (AO) for HPE 3Par enables management of data storage across two or three tiers. AO places storage regions on the appropriate tier in response to periodic analysis that AO performs.

To work with the storage in an AO group, Turbonomic:

- Discovers each Common Provisioning Group (CPG) as a disk array

In the Turbonomic user interface, these disk arrays do not host storage — They appear empty.

Turbonomic will not recommend storage moves between these disk arrays, because such moves would conflict with AO block-level placement.

- Creates a single logical pool that hosts all the datastores in an AO group

This logical pool represents the AO group, and it includes all the member CPGs. Turbonomic considers this single logical pool when it performs analysis — It can recommend moving storage into or out of the AO group. Also, Turbonomic aggregates resource capacity in this logical pool. For example, the IOPS capacity for the AO logical pool is a combination of IOPS capacity for the constituent CPGs.

You can see the AO logical pool in the Turbonomic user interface. The display name for this logical pool is the name of the AO Configuration.

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type        | Can Be Automated          | Recommendations only      |
|--------------------|---------------------------|---------------------------|
| Storage            | Provision, Resize Up/Down |                           |
| Disk Array         | Provision, Resize Up/Down |                           |
| Logical Pool       |                           | Provision, Resize Up/Down |
| Storage Controller |                           | Provision                 |

## Monitored Resources

Turbonomic monitors the following storage resources:

- Storage

- Storage Amount

The utilization of the datastore's capacity.

- Storage Provisioned

The utilization of the datastore's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency

The utilization of latency on the datastore.

- Disk Array

- Storage Amount

The utilization of the Disk Array's capacity.

- Storage Provisioned

The utilization of the Disk Array's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the disk array.

- Latency

The utilization of latency, computed from the latency of each device in the disk array.

- Logical Pool

- Storage Amount

The utilization of the logical pool's capacity.

- Storage Provisioned

The utilization of the logical pool's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the logical pool.

- Latency

The utilization of latency on the logical pool.

- Storage Controller

**Note:**

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- CPU

The utilization of the Storage Controller's allocated CPU.

- Storage Amount

The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## IBM FlashSystem

IBM® FlashSystem family comprises all-Flash storage-array platforms. The platform delivers an active-active dual controller system. Assuming a pair of controller nodes, the active-active controller pair can process I/O for a specific volume through either node.

A FlashSystem control enclosure contains multiple NVMe-attached IBM FlashCore® Modules or other self-encryption NVMe-attached SSD drives. Each control enclosure contains two identical node canisters. A node canister provides host interfaces, management interfaces, and interfaces to the control enclosure. The pair of nodes within a single enclosure is known as an input/output (I/O) group.

**Note:**

For deployments that include mirrored volumes, Turbonomic discovers the Primary copy, and uses data from that for analysis and to generate actions. It does not discover the Secondary copy.

### Prerequisites

- A service user account on the FlashSystem controller

You must configure a user account on the FlashSystem with a `Monitor` role. This user account must be dedicated to a single Turbonomic instance. To manage one FlashSystem with multiple Turbonomic instances, you must create a different user account for each Turbonomic instance.

- Spectrum Virtualize REST version 8.3.1.2 or later (8.4.2.0 or later recommended)

To take advantage of significant improvements to the Spectrum Virtualize interface, we recommend that you use Spectrum Virtualize version 8.4.2.0 or later.

### IBM FlashSystem Permissions

| Turbonomic Functionality | Required Permissions                                                                                 |
|--------------------------|------------------------------------------------------------------------------------------------------|
| Monitoring               | <ul style="list-style-type: none"> <li>• Service Account with a <code>Monitor</code> role</li> </ul> |

### Considerations

When you configure a FlashSystems target, you should consider:

- FlashSystem devices under IBM SVC management are not currently supported.
- Turbonomic does not discover mirrored volumes or child pools.
- Clusters must consist of a single enclosure.

## Adding IBM FlashSystem Targets

For FlashSystem targets, select the **Storage > IBM FlashSystem** option on the Target Configuration page. Then provide the following:

- Address  
The host name or Management IP address of the FlashSystem Cluster.
- Username  
The username for a dedicated Turbonomic user account on the FlashSystem instance.
- Password  
The password for the given user account.

## Supply Chain

IBM FlashSystem Cluster targets add Storage Controller, Disk Array, and Logical Pool entities to the supply chain. Logical Pool entities provide Storage entities, which the supply chain stitches to Hypervisor storage (for example, vCenter datastores).

The FlashSystem components that Turbonomic discovers map to supply chain entities as follows:

| FlashSystem Name                 | Supply Chain Entity | Description                                                                                                                                                         |
|----------------------------------|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Volume (VDisk)                   | Storage             | A logical disk that the system presents to attached hosts. You create Volumes from the extents that are available in the pool.                                      |
| Pool (MDiskGroup)                | Logical Pool        | An allocated amount of capacity that jointly contains all of the data for a specified set of volumes. All MDisks in a pool are split into extents of the same size. |
| MDisk (Array)                    | Disk Array          | A logical unit of physical storage. MDisks are not visible to host systems.                                                                                         |
| System (Enclosure or Controller) | Storage Controller  | A FlashSystem control enclosure.                                                                                                                                    |

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity             | Actions                                                                                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Storage Controller | <ul style="list-style-type: none"><li>• Disabled by default<ul style="list-style-type: none"><li>– Provision</li></ul></li></ul>                                                                   |
| Disk Array         | <ul style="list-style-type: none"><li>• Disabled by default<ul style="list-style-type: none"><li>– Resize</li><li>– Move</li><li>– Provision</li><li>– Start</li><li>– Suspend</li></ul></li></ul> |

| Entity          | Actions                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Logical Pool    | <ul style="list-style-type: none"> <li>• Recommend by default               <ul style="list-style-type: none"> <li>– Resize (add or remove MDisk)</li> </ul> </li> <li>• Disabled by default               <ul style="list-style-type: none"> <li>– Move</li> <li>– Provision</li> <li>– Start</li> <li>– Suspend</li> </ul> </li> </ul>                                                                                  |
| Storage         | <ul style="list-style-type: none"> <li>• Manual by default               <ul style="list-style-type: none"> <li>– Delete</li> </ul> </li> <li>• Recommend by default               <ul style="list-style-type: none"> <li>– Resize</li> <li>– Move</li> <li>– Provision</li> <li>– Start</li> </ul> </li> <li>• Disabled by default               <ul style="list-style-type: none"> <li>– Suspend</li> </ul> </li> </ul> |
| Virtual Machine | <ul style="list-style-type: none"> <li>• Recommend by default               <ul style="list-style-type: none"> <li>– Storage vMotion</li> </ul> </li> </ul>                                                                                                                                                                                                                                                               |

## Monitored Resources

To discover and monitor resources, Turbonomic uses the Spectrum Virtualize REST interface, and it securely retrieves `iostats` files from the configuration node.

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount  
The utilization of the datastore's capacity.
  - Storage Provisioned  
The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency  
The utilization of latency on the datastore.
- Logical Pool
  - Storage Amount  
The utilization of the logical pool's capacity.



- Storage Provisioned
  - The utilization of the logical pool's capacity, including overprovisioning.
- Storage Access Operations Per Second (IOPS)
  - The summation of the read and write access operations per second on the logical pool.
- Latency
  - The utilization of latency on the logical pool.
- Disk Array
  - Storage Amount
    - The utilization of the Disk Array's capacity.
  - Storage Provisioned
    - The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the disk array.
  - Latency
    - The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller
  - CPU
    - The utilization of the Storage Controller's allocated CPU.
  - Storage Amount
    - The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## NetApp

The Storage Control Module adds support for NetApp filers running the Data ONTAP operating system. NetApp storage controllers are Storage Virtual Machines that manage storage arrays. Turbonomic connects to these storage controllers to support NetApp targets in Cluster-Mode (C-Mode).

### Prerequisites

- Transport Layer Security (TLS) is enabled
- A service account Turbonomic can use to connect to the NetApp target

### Enabling TLS

Starting with version 5.4, by default Turbonomic requires Transport Layer Security (TLS) version 1.2 to establish secure communications with targets. NetApp filers have TLS disabled by default, and the latest version they support is TLSv1. If your NetApp target fails to validate on Turbonomic 5.4 or later, this is probably the cause.

If target validation fails because of TLS support, you might see validation errors with the following strings:

- No appropriate protocol

To correct this error, ensure that you have enabled the latest version of TLS that your target technology supports. If this does not resolve the issue, please contact IBM Technical Support.

- Certificates does not conform to algorithm constraints

To correct this error, refer to your NetApp documentation for instructions to generate a certification key with a length of 2048 or greater on your target server. If this does not resolve the issue, please contact IBM Technical Support.

For information about enabling TLS, see the Data ONTAP **System Administration Guide** for sections on the SSL protocol.

## Service User Account – Administrator Role

To discover and fully manage NetApp disk arrays, Turbonomic must have a service account that grants privileges to execute commands through the NetApp filer's OnTap API (ontapi). In most cases, you can create the administrator account via the NetApp OnCommand System Manager, or from the NetApp command line – For example:

```
security login create -role admin -username IBM -application ontapi -authmethod password.
```

If you prefer not to grant full administrator rights, see [Creating Restricted Service Accounts In NetApp](#)

## Adding NetApp Targets

To add a NetApp target, select the **Storage > NetApp** option on the Target Configuration page and provide the following information:

- Address  
The name or IP address of the NetApp cluster management server.
- Username/Password  
Credentials for the NetApp service user account that you have configured for Turbonomic to use.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in NetApp to those used in Turbonomic:

| NetApp Name        | Turbonomic Entity  |
|--------------------|--------------------|
| Volume             | Storage            |
| Aggregate          | Disk Array         |
| Controller / Filer | Storage Controller |

## Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory [Storage Supply Chain](#).

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type        | Can Be Automated | Recommendations only       |
|--------------------|------------------|----------------------------|
| Storage            | Move             | Provision, Resize Up       |
| Disk Array         |                  | Resize Up, Move, Provision |
| Storage Controller |                  | Provision                  |

Note that Turbonomic can automate moving a datastore to a disk array on the same storage controller, as well as moves to a disk array on a different storage controller.

## Monitored Resources

Turbonomic monitors the following storage resources:

### Note:

In NetApp environments, the storage controller shows 100% utilization when there are no more disks in a SPARE state that the storage controller can utilize in an aggregate. This does not indicate that the storage controller has no capacity.

- Storage

- Storage Amount

The utilization of the datastore's capacity.

- Storage Provisioned

The utilization of the datastore's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

### Note:

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency

The utilization of latency on the datastore.

- Disk Array

- Storage Amount

The utilization of the Disk Array's capacity.

- Storage Provisioned

The utilization of the Disk Array's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the disk array.

- Latency

The utilization of latency, computed from the latency of each device in the disk array.

- Storage Controller

### Note:

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- CPU

The utilization of the Storage Controller's allocated CPU.

- Storage Amount

The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## Restricted Service Accounts In NetApp

While Turbonomic prefers a NetApp service account with administrator rights, it is possible to create an account that has limited access, by following the steps outlined below, depending on NetApp mode.

## NetApp 9.x Restricted Service Account Setup

If you prefer to use a service account that does not have full administrator rights:

1. Log into the NetApp filer from a command shell.
2. Create a role and assign it permission to execute each of the following commands:

For example:

```
security login role create -role RoleName -cmddirname "storage aggregate show" -vserver Cluster-Name
```

The required capabilities are listed below:

- cluster identity modify
- cluster identity show
- lun create
- lun igroup create
- lun igroup modify
- lun igroup show
- lun mapping create
- lun mapping delete
- lun mapping show
- lun modify
- lun show
- network interface create
- network interface delete
- network interface modify
- network interface show
- statistics show
- storage aggregate create
- storage aggregate modify
- storage aggregate show
- storage disk show
- system controller flash-cache show
- system node modify
- system node show
- version
- volume create
- volume modify
- volume move modify
- volume move show
- volume move start
- volume qtree create
- volume qtree show
- volume show
- volume snapshot create
- volume snapshot modify
- volume snapshot show

- vserver create
  - vserver fcp nodename
  - vserver iscsi nodename
  - vserver modify
  - vserver options
  - vserver show
3. For execution privileges, execute the following commands for the given role, where Role-Name is the name of the role you are creating, and Cluster-Name identifies the cluster you want the role to affect. You must execute these commands individually to set privileges that affect each individual cluster:
    - security login role create -role Role-Name -access all -cmddirname "volume offline" -vserver Cluster-Name
    - security login role create -role Role-Name -access all -cmddirname "volume unmount" -vserver Cluster-Name
    - security login role create -role Role-Name -access all -cmddirname "volume move" -vserver Cluster-Name
    - security login role create -role Role-Name -access all -cmddirname "volume delete" -vserver Cluster-Name
  4. Create a user that will use the newly-created role.  
For example:
 

```
security login create -User-Name RoleUser -r TurbonomicRole
```
  5. Enter a password for the new user when prompted.
  6. Give the user access to the ssh and ontapi applications by using the following commands, replacing Role-Name and RoleUser with the role and user you created:
 

```
security login create -role Role-Name -username RoleUser -application ontapi -authmethod password
```

```
security login create -role Role-Name -username RoleUser -application ssh -authmethod password
```

## NetApp C-Mode Restricted Service Account Setup

If you prefer to use a service account that does not have full administrator rights:

1. Log into the NetApp filer from a command shell.
2. Create a role and assign it permission to execute each of the following commands:
  - aggr-get-iter
  - igroup-get-iter
  - cluster-identity-get
  - lun-map-get-iter
  - net-interface-get-iter
  - storage-disk-get-iter
  - system-get-node-info-iter
  - volume-get-iter
  - vserver-get-iter
  - fcp-node-get-name
  - flash-device-get-iter
  - iscsi-node-get-name

- options-list-info
- qtree-list-iter
- system-get-version
- lun-get-iter
- snapshot-get-iter
- perf-object-get-instances
- volume-get-iter
- volume-move-get-iter
- volume-move-start

For example, to enable volume offline, execute the following:

```
security login role create -role Role-Name -access all -cmddirname "volume
offline" -vserver <cluster_name>
```

3. Create a user based on the role you create.

Give the user access to the ssh and ontapi applications. For example:

```
security login create -role Role-Name -username User-Name -application ontapi -authmethod
password
```

## Pure Storage

Turbonomic supports management of the following Pure Storage technologies:

- FlashArray//C
- FlashArray//X

The following technologies are not supported:

- FlashBlade

Because of the improved performance of Pure Storage arrays, Turbonomic intelligently moves more demanding workloads to Flash-based datastores. Turbonomic analysis is also able to incorporate Pure Storage de-duplication and compression when recommending actions.

### Prerequisites

- A service account Turbonomic can use to connect to the FlashArray

This account needs privileges to execute commands through the Pure Storage API — Typically the default `pureuser` administrative account.

### Adding Pure Storage Targets

To add a Pure Storage target, select the **Storage > Pure Storage** option on the Target Configuration page and provide the following information:

- Address

The name or IP address of the Pure Storage FlashArray.

- Username/Password

Credentials for the service account Turbonomic can use to connect to the FlashArray. The Username must not contain the domain. For example, `Username=jjsmith` is correct, while `Username=myDomain\jjsmith` will result in a failure to validate.

- Secure connection

When checked, uses SSL to connect to the Pure target. Most Pure installations do not accept insecure connections. If you receive an error when adding the target with secure connections disabled, try re-adding with this option enabled.

After validating the new target, Turbonomic discovers the connected storage entities. This table compares terms used in Pure to those used in Turbonomic:

| Pure Name   | Turbonomic Entity  |
|-------------|--------------------|
| Volume      | Storage            |
| Shelf Array | Disk Array         |
| Controller  | Storage Controller |

## Supply Chain

Storage targets (storage controllers) add Storage Controller and Disk Array entities to the supply chain. Disk Array entities then host Storage entities (datastores). For a visual representation, see the introductory [Storage Supply Chain](#).

## Supported Actions

For each discovered entity, Turbonomic can execute or recommend certain actions, as outlined below.

| Entity Type        | Can Be Automated | Recommendations only |
|--------------------|------------------|----------------------|
| Storage            |                  | Resize Up            |
| Disk Array         |                  |                      |
| Storage Controller |                  | Provision            |

Pure Storage assigns all the disks managed by a storage controller to a single array, with a fixed form-factor. There are no actions to perform for an array — For example, there is no action to move a disk array from one storage controller to another. Likewise, there are no actions to move or provision volumes because of the fixed form-factor.

## Monitored Resources

Turbonomic monitors the following storage resources:

- Storage
  - Storage Amount
    - The utilization of the datastore's capacity.
  - Storage Provisioned
    - The utilization of the datastore's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the datastore.

### Note:

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency

- The utilization of latency on the datastore.
- Disk Array
  - Storage Amount
    - The utilization of the Disk Array's capacity.
  - Storage Provisioned
    - The utilization of the Disk Array's capacity, including overprovisioning.
  - Storage Access Operations Per Second (IOPS)
    - The summation of the read and write access operations per second on the disk array.
  - Latency
    - The utilization of latency, computed from the latency of each device in the disk array.
- Storage Controller
 

**Note:**

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

  - CPU
    - The utilization of the Storage Controller's allocated CPU.
  - Storage Amount
    - The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

## Virtual Desktop Infrastructure Targets

---

A virtual desktop infrastructure target is a service that delivers digital workspaces to multiple application users through a single portal. When you connect Turbonomic to VDI targets, it monitors the performance and resource consumption of the business users utilizing the virtual desktops and the virtual desktop infrastructure itself. Turbonomic then performs Application Resource Management (ARM) using the enhanced data to assure application performance while also utilizing resources as efficiently as possible.

As part of this process, Turbonomic will use this information to provide deeper insight into the state of the hardware, and information related to the users and virtual desktops running in your environment.

### Monitored Resources

Turbonomic monitors the following resources for the VDI supply chain:

- Business User
  - ImageCPU
    - CPU utilization, as a percentage of CPU capacity for the user's desktop image or images.
  - ImageMem
    - Memory utilization, as a percentage of Memory capacity for the user's desktop image or images.
  - ImageStorage
    - Storage utilization, as a percentage of storage capacity for the user's desktop image or images.
- Desktop Pool
  - Pool CPU
    - The CPU available to the pool that is in use by active sessions.
  - Pool Memory



The memory available to the pool that is in use by active sessions.

- Pool Storage

The storage capacity available to the pool that is in use by active sessions.

- Active Sessions

How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.

- Total Sessions

How many active and disconnected (non-terminated) sessions are on the pool, as a percentage of the pool's capacity.

## Horizon

VMware Horizon provides end users access to all of their virtual desktops, applications, and online services through a single digital workspace.

With the information provided by Horizon, Turbonomic efficiently utilizes the VDI infrastructure to avoid inappropriately placed or sized desktops, ensure users are matched to desktop images that meet their demand, while gaining valuable insight into the interconnected nature of your environment.

### Prerequisites

- Horizon 7.0 or higher
- The version of the vCenter target associated to the Horizon target must be 5.0+

### Adding Horizon Targets

#### Note:

For each Horizon target, you must add only one connection server servicing that site. Turbonomic will use the default port of 443. Adding multiple connection servers for the same site will result in inconsistent and/or duplicate actions.

To add a Horizon target, select the **Virtual Desktop Infrastructure > Horizon** option on the Target Configuration page and provide the following information:

- Address

The IP address of the Horizon connection server.

- Username

Username for the user account Turbonomic will use to connect to the Horizon target. This account must be an Administrator (Read-Only) account.

- Password

Password for the user account Turbonomic will use to connect to the Horizon target.

- vCenter Username

Username for the user account Turbonomic will use to connect to the vCenter target related to Horizon. Leave this blank if the Horizon and vCenter credentials are identical.

- vCenter Password

Password for the user account Turbonomic will use to connect to the vCenter target related to Horizon. Leave this blank if the Horizon and vCenter credentials are identical.

- LDAP Server

The address of the specific LDAP server Turbonomic will use in conjunction with the Horizon target. By default, Turbonomic will do a lookup based on the domain entered in the Username and select a healthy DNS server from the response.

- LDAP Server Port

The port Turbonomic will use to connect to the specified LDAP server. By default, Turbonomic will use port 389 for unsecure connections, and port 636 for secure connections.

- Secure LDAP Connection

When checked, Turbonomic will use SSL to communicate to the LDAP Server

- LDAP Username

The username Turbonomic will use to connect to the LDAP Server, in the user principal name format to authenticate in the directory service through Kerberos. For example, `user.name@domain.com`

- LDAP Password

The password Turbonomic will use to connect to the LDAP Server

## Actions

Turbonomic recommends actions for the various entities of the Horizon target as follows:

- Desktop Pool

- Turbonomic does not recommend actions to perform on a desktop pool. Instead, it recommends actions to perform on the business users running active sessions in the pool.

- View Pod

- Turbonomic does not recommend actions to perform on a view pod. Instead, it recommends actions to perform on the business users running active sessions in the view pod.

- Business User

- Move Business User due to image congestion
- Move Business User due to desktop pool congestion

Under some circumstances, you can have Business Users who need larger desktop images. This appears as users with high utilization of the image resources. In this case, Turbonomic can recommend moving the Business Users to a different desktop pool that serves up larger images.

To support moving Business Users, you must create a placement policy that merges desktop pools. Be sure to merge only desktop pools that are *similarly configured* – they should run the same operating system and applications, and differ only in allocated memory and/or CPU.

To merge desktop pools:

1. Create a new placement policy.
2. Choose **Merge** as the policy type.
3. For the consumer type to merge, choose **Desktop Pool**.
4. Choose the pools that you want to merge.
5. Save the policy.

For more information, see [Creating Placement Policies](#).

## Monitored Resources

Turbonomic monitors the following commodities of the Horizon target:

- Business User

- ImageCPU

CPU utilization, as a percentage of CPU capacity for the user's desktop image or images.

- ImageMem

Memory utilization, as a percentage of Memory capacity for the user's desktop image or images.

- ImageStorage
  - Storage utilization, as a percentage of storage capacity for the user's desktop image or images.
- Desktop Pool
  - Pool CPU
    - The CPU available to the pool that is in use by active sessions.
  - Pool Memory
    - The memory available to the pool that is in use by active sessions.
  - Pool Storage
    - The storage capacity available to the pool that is in use by active sessions.
  - Active Sessions
    - How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.
  - Total Sessions
    - How many active and disconnected (non-terminated) sessions are on the pool, as a percentage of the pool's capacity.
- View Pod
  - Active Sessions
    - How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.
  - Total Sessions
    - How many active and disconnected (non-terminated) sessions are on the pool, as a percentage of the pool's capacity.

## Appendix – Target Configuration

---

This appendix contains topics that are related to configuring Turbonomic targets.

### Cisco Unified Computing System

#### UCS Blade Provisioning

When managing a UCS fabric target, Turbonomic can provision any blade servers that are installed in a chassis but not currently in operation. If the workload demands more physical compute resources, Turbonomic can automatically direct UCS to provision a blade, or it can recommend that you provision a blade and you can execute the action from the To Do list. To enable this capability, you must perform two basic steps:

- Configure the way UCS and vCenter Server manage information as blades are provisioned

To enable Turbonomic to perform automatic provisioning of UCS blades, you must configure UCS to work with vCenter Server as they both manage resources such as server pools, policies, and Service Profile Templates. This ensures that as Turbonomic directs the UCS Manager to provision a new blade, vCenter Server will recognize that the new physical host is available. Once vCenter Server can recognize the new blade, Turbonomic can direct vCenter Server to move workloads onto the server.

Turbonomic provisions new blades based on the service profiles of operating blades. To enable this, the configuration must include Service Profile Templates, and the operating blades must be bound to these templates.

For information about configuration that enables automated provisioning of blades, see the Cisco Communities post, “UCS PowerTool and VMware PowerCLI automated management of Auto-deploy” at the following location:

[https://communities.cisco.com/community/technology/datacenter/ucs\\_management/cisco\\_ucs\\_developed\\_integrations/blog/2013/09/16/ucs-powertool-and-vmware-powercli-automated-management-of-auto-deploy](https://communities.cisco.com/community/technology/datacenter/ucs_management/cisco_ucs_developed_integrations/blog/2013/09/16/ucs-powertool-and-vmware-powercli-automated-management-of-auto-deploy)

This post includes a video that shows “a joint PowerShell integration utilizing both Cisco UCS PowerTool and VMware PowerCLI.” You can also download the scripts from this post and modify them as necessary for your deployment.

- Set the Host Provision action to Automate or Manual for the blade servers. By default, Turbonomic sets the Host Provision action to Recommend.

For any hosts other than blade servers managed by UCS, Turbonomic cannot provision hosts automatically or manually. Instead, it recommends that you provision a host, and you then install the physical machine and wire it to the network.

In a UCS deployment you can have blade servers installed in the chassis and ready to provision. In that case, Turbonomic can direct UCS to provision a new blade to meet workload demands. For these servers, you can set the Host Provision action to Automatic or Manual.

**Note:**

It's important that you only set Automatic or Manual host provisioning to UCS blades. If you set Host Provision to Automatic for other types of hosts, Turbonomic will attempt to perform the action and fail. As a result, you might never see the recommendation to provision a new host of that type.

Turbonomic groups blade servers by chassis. To restrict Automatic or Manual settings to blade servers, use this group. You can set the action acceptance mode for all blade servers in your environment, or you can set the mode differently for individual chassis.

## Enabling Windows Remote Management

Turbonomic communicates with your Hyper-V servers using Web Services Management (WS-Management), which is implemented on Microsoft platforms using Windows Remote Management (WinRM). The following steps show how to enable WinRM on a single host, using the command line.

1. Ensure Windows Firewall is running on the host.

For you to configure WinRM successfully, Windows Firewall must be running on the host. For more information, see the Microsoft Knowledge Base article #2004640 (<http://support.microsoft.com/kb/2004640>).

2. Set up an SPN for the host machine.

The machine must have an SPN of the form, `protocol/host_address`. For example, `WSMAN/10.99.9.2`.

To get a list of SPNs for the machine, execute the following in the command window:

```
setspn -l <vmm-server-name>
```

If there is no valid SPN in the list, create one by running the command:

```
setspn -A protocol/host-address:port where port is optional
```

For example, `setspn -A WSMAN/10.99.9.2:VMM-02`

3. Set up the Windows Remote Management (WinRM) service to run on startup.

Run the `quickconfig` utility to set up the WinRM service. The `quickconfig` utility:

- Configures the WinRM service to auto-start
- Configures basic authentication and disables unencrypted traffic
- Creates a firewall exception for the current user profile

- Configures a listener for HTTP and HTTPS on any IP address
- Enables remote shell access

To run `quickconfig`, log into a command window as Administrator on the host machine. Then execute the following commands:

```
winrm quickconfig
```

Enter `y` to accept the `quickconfig` changes

#### 4. Set permissions on the host machine.

Execute the following commands in the command window to modify the settings made by `quickconfig`:

- To set the memory capacity for remote shells:

```
winrm set winrm/config/winrs @{MaxMemoryPerShellMB="1024"}
```

- To set up an unsecured HTTP connection:

```
winrm set winrm/config/service @{AllowUnencrypted="true"}
```

```
winrm set winrm/config/service/Auth @{Basic="true"}
```

These steps showed you how to enable WinRM for a single host. Some users find the following methods useful for enabling WinRM on multiple hosts:

- [EnablingWinRmViaGlobal Policy Objects](#)
- [EnablingWinRMViaPowerShell](#)

## Enabling WinRM Via Global Policy Objects

You can configure WinRM for all of your Hyper-V targets by creating and linking a Global Policy Object (GPO) within the Hyper-V domain and applying the GPO to all servers.

Follow the steps below to enable Windows Remote Management (WinRM) for your Hyper-V targets.

1. On the AD domain controller, open the Group Policy Management Console (GPMC). If the GPMC is not installed, see <https://technet.microsoft.com/en-us/library/cc725932.aspx>.
2. Create a new Global Policy Object:
  - a. In the GPMC tree, right-click **Group Policy Objects** within the domain containing your Hyper-V servers.
  - b. Choose **Create a GPO in this domain**, and link it here.
  - c. Enter a name for the new GPO and click **OK**.
3. Specify the computers that need access:
  - a. Select the new GPO from the tree.
  - b. On the **Scope** tab, under **Security Filtering**, specify the computer or group of computers you want to grant access. Make sure you include all of your Hyper-V targets.
4. Right-click the new GPO and choose **Edit** to open the Group Policy Management Editor.
5. Configure the WinRM Service:
  - a. In the Group Policy Management Editor, select **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Management (WinRM) > WinRM Service**.
  - b. Double-click each of following settings and configure as specified:

|                                                                                                                                  |                           |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| Allow automatic configuration of listeners ("Allow remote server management through WinRM" on older versions of Windows Server): | Enabled<br>IPv4 filter: * |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------|

|                             |         |
|-----------------------------|---------|
| Allow Basic authentication: | Enabled |
| Allow unencrypted traffic:  | Enabled |

6. Configure the WinRM service to run automatically:

- a. In the Group Policy Management Editor, expand **Computer Configuration > Preferences > Control Panel Settings**.
- b. Under Control Panel Settings, right-click Services and choose **New > Service**.
- c. In the New Service Properties window, configure the following settings:

|                 |               |
|-----------------|---------------|
| Startup:        | Automatic     |
| Service name:   | WinRM         |
| Service option: | Service start |

7. Enable Windows Remote Shell:

- a. In the Group Policy Management Editor, select **Computer Configuration > Policies > Administrative Templates > Windows Components > Windows Remote Shell**.
- b. Double-click the following setting and configure as specified:

|                            |         |
|----------------------------|---------|
| Allow Remote Shell Access: | Enabled |
|----------------------------|---------|

8. Add a Windows Firewall exception:

- a. In the Group Policy Management Editor, open **Computer Configuration > Windows Settings > Security Settings > Windows Firewall > Windows Firewall**.
- b. Under Windows Firewall, right-click **Inbound Rules** and choose **New > Rule**.
- c. In the New Inbound Rule Wizard, select **Predefined: Windows Remote Management and Allow the connection**.

The new group policy will be applied during the next policy process update. To apply the new policy immediately, execute the following command at a Powershell prompt:

```
gpupdate /force
```

## Enabling WinRM Via PowerShell

Using PsExec, you can run quickconfig on all your Hyper-V servers and change the default settings remotely. PsExec is a component of PsTools, which you can download from <https://technet.microsoft.com/en-us/sysinternals/bb897553.aspx>.

1. Create a text file containing the Hyper-V host names, for example:

```
hp-vx485
```

```
hp-vx486
```

2. Since IBM requires changes to the default quickconfig settings, create a batch file containing the following command:

```
@echo off Powershell.exe Set-WSManQuickConfig -Force Powershell.exe Set-Item WSMAN:\localhost\Shell\MaxMemoryPerShellMB 1024
```

### Note:

If you are connecting via HTTP, you must include the following command:

```
Powershell.exe Set-Item WSMAN:\localhost\Service\AllowUnencrypted -Value $True
```

3. Use PsExec to enable WinRM on the remote servers:

```
.\PsExec.exe @<hosts_file_path> -u <username> -p <password> -c
<batch_file_path>
```

**Note:**

If you get an error message when executing this command, add the -h option (.\PsExec.exe -h).

## Secure Setup of WSMAN

Turbonomic provides a secure option for Hyper-V/VMM Targets which requires that WSMAN be set up securely. Use PowerShell to generate a self-signed certificate, and create an HTTPS WinRM listener.

**Note:**

For clustered Hyper-V targets, you do not need to create a listener on each host. Only create a listener on the host that is being added to the "Address" field in the Target Configuration.

To enable secure WSMAN on your Hyper-V host:

1. Generate a self-signed certificate using the following command:

```
New-SelfSignedCertificate -CertstoreLocation Cert:\LocalMachine\My -DnsName
"myhost.example.org"
```

2. Find the thumbprint for the certificate for the host:

```
Get-childitem cert:\LocalMachine\My
```

3. Create an HTTPS WinRM listener for the host with the thumbprint you've found:

```
winrm create winrm/config/Listener?
Address=**+Transport=HTTPS '@{Hostname="myhost.example.org";
CertificateThumbprint="THUMBPRINT_YOU_FOUND"}'
```

4. Verify the presence of configured listeners:

```
Get-WSManInstance -ResourceURI winrm/config/listener -Enumerate
```

## Port Configuration

To support communication between Turbonomic and your intended targets, Turbonomic must be able to reach the targets via the following ports:

### Applications and Databases

| Target      | Ports      |
|-------------|------------|
| AppDynamics | 80, 443    |
| JBoss       | 9990       |
| MySQL       | 3306       |
| Oracle      | 1521       |
| SQL Server  | 1434, 1433 |
| Tomcat      | 1009       |
| WebLogic    | 7001       |

| Target    | Ports |
|-----------|-------|
| WebSphere | 8880  |

### Fabric and Network

| Target    | Ports   |
|-----------|---------|
| Cisco UCS | 80, 443 |

### Hyperconverged

| Target  | Ports |
|---------|-------|
| Nutanix | 9440  |

### Hypervisors

| Target               | Ports      |
|----------------------|------------|
| Hyper-V              | 5985, 5986 |
| Hyper-V (Kubernetes) | 88         |
| vCenter (Monitoring) | 443        |
| vCenter (Tags)       | 10443      |
| vCenter (Kubernetes) | 88         |

### Private Cloud

| Target        | Ports      |
|---------------|------------|
| Microsoft VMM | 5985, 5986 |

### Storage

| Target          | Ports            |
|-----------------|------------------|
| EMC VMAX        | 5988, 5989       |
| HPE 3PAR        | 5988, 5989, 8080 |
| IBM FlashSystem | 7443             |
| NetApp          | 80, 443          |
| Pure Storage    | 80, 443          |



## Virtual Desktop Infrastructure

| Target         | Ports |
|----------------|-------|
| VMware Horizon | 443   |

## AWS Target IAM Role Requirements

This appendix guides you through the steps to configure the Turbonomic AWS Mediation pods to leverage IAM roles. To do so, you must leverage the ability to provide fine grained IAM role support through a Service Account, and Kubernetes cluster configurations that support an OIDC provider and webhook method.

### Note:

The Turbonomic SaaS offering also supports adding AWS targets via IAM roles. See [Turbonomic SaaS IAM Role Setup](#).

## Prerequisites

Follow the steps below to configure the AWS Mediation pods to leverage IAM roles:

1. Follow the instructions provided by Amazon EKS, Red Hat OpenShift, or GKE to make sure you have the required configurations to support the Web Identity provider method, leveraging the AWS webhook and an OIDC provider.
  - [Introducing EKS Granular IAM Roles for Pods via Service Accounts and OIDC Providers](#)
  - [Overview of setting up IAM Role for EKS WebIdentity Provider](#)
  - [Creating a Service Account with an IAM Role: EKS example](#)
  - [Red Hat OpenShift support for Granular IAM Roles](#)
  - GKE and IAM roles described in this [blog using gtoken to inject the token into the pod](#)
  - [AWS IAM Role Permissions and Trust Relationships](#)

### Note:

Follow the instructions from AWS in the links above to set up your cluster's OIDC provider as an IAM Identity Provider (Web Identity Provider - OIDC URL) in the AWS account that you will be targeting.

2. Configure a Kubernetes Service Account in the Turbonomic namespace that will assume an IAM role.
  - (Best Practice) Manually create a separate Service Account for the AWS Mediation pods to use. You must then modify the Custom Resource YAML to specify this Service Account to the AWS Mediation components. For example:

```
spec:
 mediation-aws:
 serviceAccountName: t8c-iam-role
 mediation-awsbilling:
 serviceAccountName: t8c-iam-role
 mediation-awscost:
 serviceAccountName: t8c-iam-role
```

After you update the Custom Resource YAML, apply the updated Custom Resource and ensure the three Mediation pods restarted.

- Use the Turbonomic default Service Account with which the AWS Mediation pods are running.

### Note:

Unless specified in the Turbonomic Custom Resource, the AWS Mediation pods will run with the default service account called "default" in the namespace. If you modify the default "default" account, there is nothing more to do except restart the AWS Mediation pods.

3. Configure the IAM role in AWS.

- Turbonomic AWS IAM role requirements, including cross account access if required, are described [here](#). IAM policy definition must use the `sts:AssumeRoleWithWebIdentity` role.
- Using the Web identity provider set up, and the Service Account you will use, update the Trust Relationships in the IAM role. See the instructions [here](#).
- The IAM role for Turbonomic to target your AWS account requires the following policies:

```
AmazonRDSReadOnlyAccess
AmazonEC2ReadOnlyAccess
AmazonS3ReadOnlyAccess
AWSOrganizationsReadOnlyAccess
```

4. Annotate the Service Account you will use with the IAM role. See [this article](#) for an example.

```
apiVersion: v1
kind: ServiceAccount
#use the name of the SA that will contain the annotation
name: default
metadata:
 annotations:
 eks.amazonaws.com/role-arn: arn:aws:iam::<AWS_ACCOUNT_ID>:role/<IAM_ROLE_NAME>
```

5. Complete the Turbonomic AWS Account Target setup by going to the Turbonomic UI to configure the AWS Account targets using an IAM role. See [Amazon Web Services](#).

---

## Chapter 8. User Interface Reference

IBM Turbonomic is a solution that assures application performance for any workload running in any virtualized or cloud environment.

After installing or setting up the product, you can see the results of Turbonomic analysis — actions to perform that directly improve your datacenter state — within 15 to 30 minutes.

While it's true that setup is straightforward, the Turbonomic platform is rich in its coverage. Analysis takes your full stack into account, and Turbonomic recommends actions before alarms go off and the situation becomes critical. The user interface for such a product is necessarily rich with high-level information as well as fine details. You can use this interface and drill down to specific details that clarify the actions Turbonomic recommends.

IBM also understands the need to define business rules in your environment. For example, you might need to ensure that certain applications have exclusive access to “golden” storage, while others can use less expensive resources. Or, you might want to ensure that certain workloads resize at scheduled times. Turbonomic automation supports this kind of business rule, as well as HA, affinity, discount purchase profiles, and many other rules that you expect to set up in a modern datacenter. The user interface includes tools to configure Turbonomic so that action recommendations respect the needs within your environment.

This reference material describes Turbonomic and its user interface. If you have any questions about what you see in the user interface, or how to use the tools that Turbonomic offers, we hope you will find this material useful.

---

### Entity Types - Applications

The supply chain strongly emphasizes our application-driven approach to managing your infrastructure. By showing the entity types that make up your applications at the top of the hierarchy, it is easier for you to see the health of your environment and evaluate actions from the perspective that matters — Application Performance.



## Business Application

A Business Application is a logical grouping of [Business Transactions](#), [Services](#), [Application Components](#), and other elements of the application model that work together to compose a complete application as end users would view it. For example, a mobile banking app is a Business Application with a *Business Transaction* that facilitates payments, a *Service* within the Business Transaction that records payment information, and underlying *Application Components* (such as JVMs) that enable the Service to perform its functions.



You can monitor overall performance, make resourcing decisions, and set policies in the context of your Business Applications.

### Note:

The supply chain also models Turbonomic as a Business Application so you can monitor the performance of the product. For details, see [Turbonomic as a Business Application](#).

## Synopsis

| Synopsis       |                                                                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:        | Business Applications have unlimited budget.                                                                                                                                                                                                                                                                                                             |
| Provides:      | The complete application to end users                                                                                                                                                                                                                                                                                                                    |
| Consumes from: | Business Transactions, Services, Application Components, Database Servers, and the underlying nodes                                                                                                                                                                                                                                                      |
| Discovery:     | <p>Turbonomic discovers the following:</p> <ul style="list-style-type: none"> <li>• AppDynamics Business Applications</li> <li>• Dynatrace Applications</li> </ul> <p>If you do not have these targets, you can create your own Business Applications using the Application Topology feature. For details, see <a href="#">Application Topology</a>.</p> |

## Monitored Resources

Turbonomic monitors the following:

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

The **Response Time** and **Transaction** charts for a Business Application show average and peak/low values over time. You can gauge performance against the given SLOs. By default, Turbonomic estimates SLOs based on monitored values. You can set your own SLO values in policies.

## Actions

### None

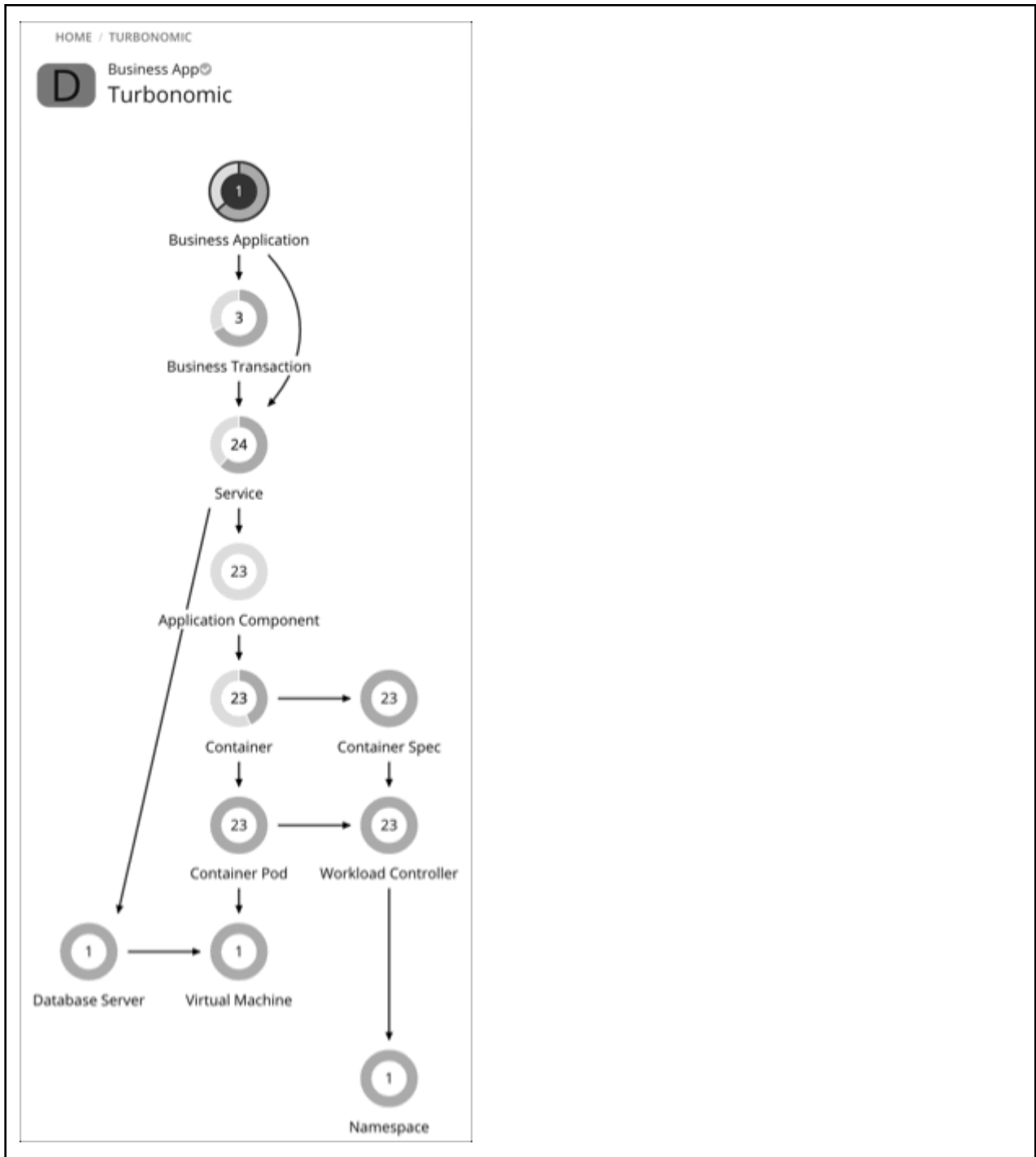
Turbonomic does not recommend actions for a Business Application, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Application lists these actions, thus providing visibility into the risks that have a direct impact on the Business Application's performance.

## Turbonomic as a Business Application

The supply chain models Turbonomic as a Business Application. When you scope to the Turbonomic entity, you can see the Business Transactions, Services, and Application Components that make up the product, the underlying nodes (such as the product's VM and database server), as well as performance metrics.

### Note:

After deploying Turbonomic, it might take at least ten minutes before it appears in the supply chain as a Business Application.



Key Business Transactions, Services, and Application Components work together to:

- Analyze your environment continuously
- Recommend actions to assure performance and reduce cloud costs
- Keep the supply chain up-to-date
- Run plans against the current environment

The Pending Actions chart at every level of the application model shows actions for the underlying Application Components and nodes, thus providing visibility into the risks that have a direct impact on the product's performance.

## Business Application Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

#### None

Turbonomic does not recommend actions for a Business Application, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Application lists these actions, thus providing visibility into the risks that have a direct impact on the Business Application's performance.

### Transaction SLO

Enable this SLO if you are monitoring performance through your Business Applications.

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                    |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

Transaction SLO determines the upper limit for acceptable transactions per second. When the number of transactions reaches the given value, Turbonomic sets the risk index to 100%.

### Response Time SLO

Enable this SLO if you are monitoring performance through your Business Applications.

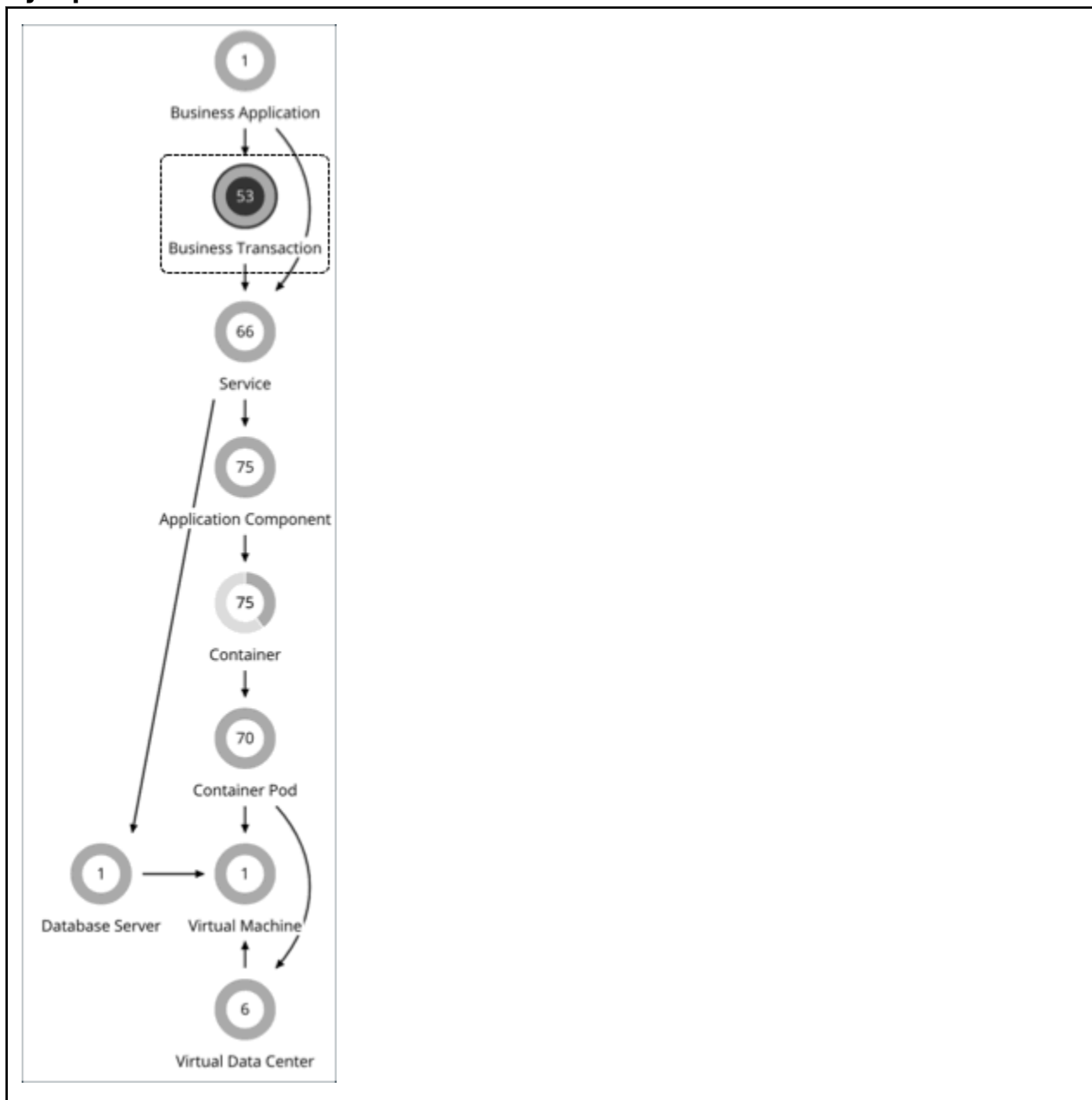
| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                      |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

Response time SLO determines the upper limit for acceptable response time (in milliseconds). If response time reaches the given value, Turbonomic sets the risk index to 100%.

## Business Transaction

A Business Transaction represents a capability within your Business Application that fulfills a response to a user-initiated request. Its performance directly impacts user experience. You can monitor performance as experienced by your end users in the context of Business Transactions. For more information, see [Business Application](#).

## Synopsis



### Synopsis

|                |                                                                                                                                                                                                                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:        | Business Transactions have unlimited budget.                                                                                                                                                                                                                                                                                                                |
| Provides:      | Response time and transactions to Business Applications                                                                                                                                                                                                                                                                                                     |
| Consumes from: | <a href="#">Services</a> , <a href="#">Application Components</a> , <a href="#">Database Servers</a> , and the underlying nodes                                                                                                                                                                                                                             |
| Discovery:     | <p>Turbonomic discovers the following:</p> <ul style="list-style-type: none"> <li>• AppDynamics Business Transactions</li> <li>• NewRelic Key Transactions</li> </ul> <p>If you do not have these targets, you can create your own Business Transactions using the Application Topology feature. For details, see <a href="#">Application Topology</a>.</p> |



## Monitored Resources

Turbonomic monitors the following:

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

The **Response Time** and **Transaction** charts for a Business Transaction show average and peak/low values over time. You can gauge performance against the given SLOs. By default, Turbonomic estimates SLOs based on monitored values. You can set your own SLO values in policies.

## Actions

### None

Turbonomic does not recommend actions for a Business Transaction, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Transaction lists these actions, thus providing visibility into the risks that have a direct impact on the Business Transaction's performance.

## Business Transaction Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

### None

Turbonomic does not recommend actions for a Business Transaction, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Transaction lists these actions, thus providing visibility into the risks that have a direct impact on the Business Transaction's performance.

## Transaction SLO

Enable this SLO if you are monitoring performance through your Business Transactions.

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                    |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

Transaction SLO determines the upper limit for acceptable transactions per second. When the number of transactions reaches the given value, Turbonomic sets the risk index to 100%.

## Response Time SLO

Enable this SLO if you are monitoring performance through your Business Transactions.

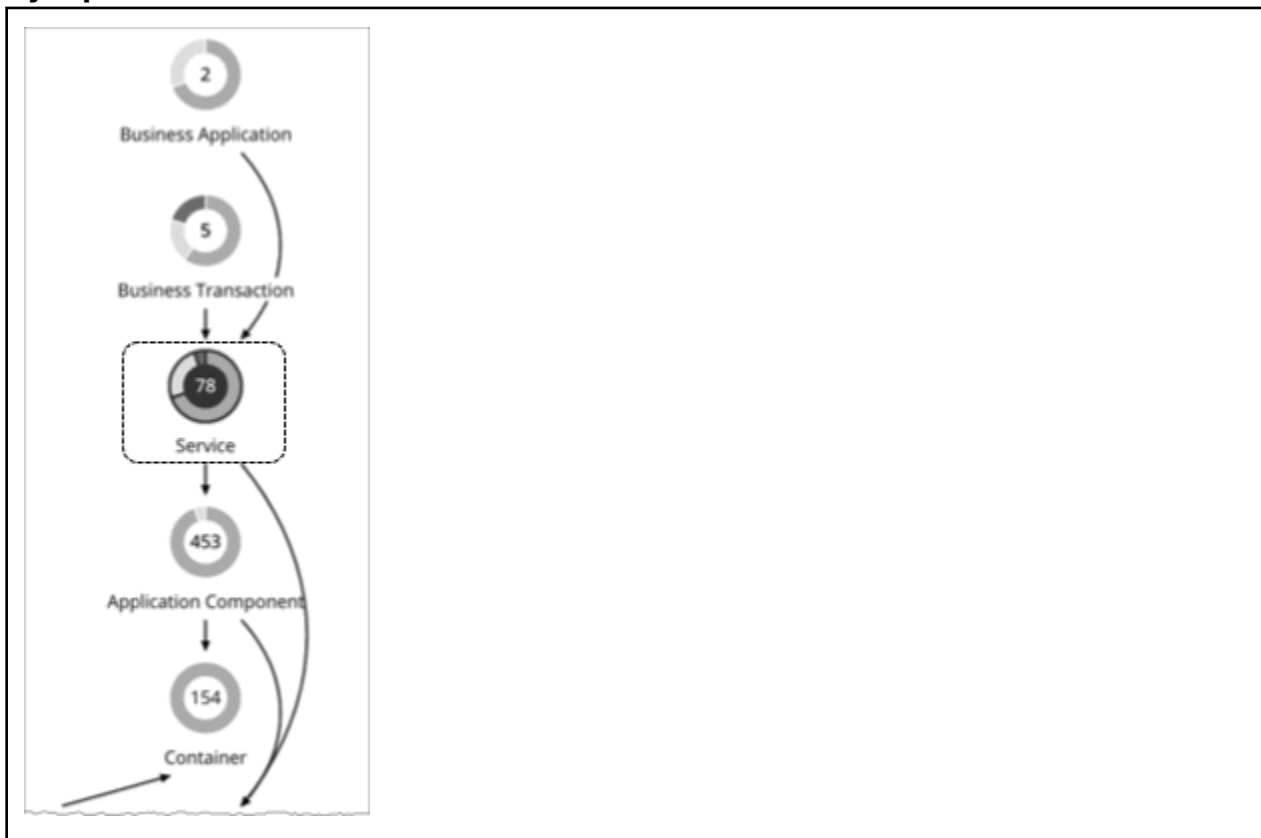
| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                      |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

Response time SLO determines the upper limit for acceptable response time (in milliseconds). If response time reaches the given value, Turbonomic sets the risk index to 100%.

## Service

A Service in the supply chain represents one or several Application Components that perform a defined, measurable function as part of an internal or user-initiated request. Its performance is key to understanding application performance, but only indirectly impacts user experience. You can measure performance as experienced internal to the Business Application in the context of Services.

## Synopsis



### Synopsis

Provides:

Response time and transactions to Business Transactions and Business Applications

## Synopsis

Consumes from:

Application Components, Database Servers, and the underlying nodes

Discovery:

- APM targets

Turbonomic discovers the following:

- AppDynamics Tiers
- Dynatrace Services
- Instana Services
- NewRelic APM Applications / NewRelic Services (New Relic ONE)

**Note:**

If you do not have an APM target, you can create your own Services using the Application Topology feature. For details, see [Application Topology](#).

- Cloud native targets

Turbonomic discovers Kubernetes Services through the Kubeturbo pod that you have deployed in your environment.

A Kubernetes Service is a logical set of pods that represents a given application. The Service exposes a single entry point for the application process. While the pods that comprise the service are ephemeral, the service is persistent. The Service entity also gives historical tracking of the number of replicas that run to support the Service.

## Monitored Resources

Turbonomic monitors the following resources for both non-Kubernetes and Kubernetes Services:

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Transaction

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

The **Response Time** and **Transaction** charts for a Service show average and peak/low values over time. You can gauge performance against the given SLOs. By default, Turbonomic estimates SLOs based on monitored values. You can set your own SLO values in policies.

## Actions for non-Kubernetes Services

Turbonomic does not recommend actions for non-Kubernetes Services, but it does recommend actions for the underlying Application Components and nodes. The Pending Actions chart for Services list these actions, thus providing visibility into the risks that have a direct impact on their performance.

## Actions for Kubernetes Services

### Provision or Suspend

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

For example, when current response time for an application is in direct violation of SLO, Turbonomic will recommend provisioning pods to improve response time. When you examine a pending action to provision pods, you will see *Response Time Congestion* as the reason for the action.

| Container Pod Name               | Container Cluster     | Namespace  | Risk                     |
|----------------------------------|-----------------------|------------|--------------------------|
| robot-shop/ratings-857bd6d9c4-5v | Kubernetes-endre-dc11 | robot-shop | Response Time Congestion |
| robot-shop/ratings-857bd6d9c4-5v | Kubernetes-endre-dc11 | robot-shop | Response Time Congestion |
| robot-shop/user-6ccb589cbd-4hbk  | Kubernetes-endre-dc11 | robot-shop | Response Time Congestion |

If applications can meet response time SLOs using less resources, Turbonomic will recommend suspending pods to improve infrastructure efficiency.

| Container Pod Name               | Container Cluster      | Namespace | Risk                              |
|----------------------------------|------------------------|-----------|-----------------------------------|
| demoapp/twitter-cass-api-5bd5898 | Kubernetes-ccp-testbed | demoapp   | Improve infrastructure efficiency |
| demoapp/twitter-cass-tweet-66588 | Kubernetes-ccp-testbed | demoapp   | Improve infrastructure efficiency |

### Action Generation Requirements

Turbonomic generates actions for Kubernetes Services under the following conditions:

- Services are discovered by the [Kubeturbo](#) pod that you have deployed in your environment.
- Application performance metrics for Services are collected via the [Instana](#) target or the [Prometurbo](#) metrics server.

Prometurbo collects application performance metrics from [Prometheus](#), and then exposes the applications and metrics in JSON format via REST API. The [DIF](#) (Data Ingestion Framework) probe accesses the REST API and converts the JSON output to a DTO consumed by Turbonomic.

Deploy Prometurbo and create the necessary Custom Resources (CRs) to enable the collection of performance metrics. CRs include [PrometheusQueryMapping](#) and [PrometheusServerConfig](#). With CRs, you can quickly update the configurations used for metric collection without needing to restart Prometurbo. Before installing the CRs, be sure to install the Custom Resource Definitions (CRDs) used to validate the CRs.

For details about Prometurbo deployment and CRs/CRDs, visit this [page](#).

- You have created policies for the Services (see the next section).

### Creating Policies for Kubernetes Services

You can create policies from two places.

- Turbonomic user interface

Navigate to **Settings > Policies**, and then click **New Automation Policy > Service**. The next section discusses how to configure the policy correctly.

The policy should be similar to the following:

The screenshot shows the 'Configure Service Policy' interface. At the top, there's a title bar with a back arrow and a close button. Below it, the 'NAME' field contains 'Policy\_A'. The 'SCOPE' section shows 'AH-Service\_GP' selected. The 'POLICY SCHEDULE' section is collapsed. The 'AUTOMATION AND ORCHESTRATION' section is expanded, showing 'Action Acceptance: Manual'. The 'OPERATIONAL CONSTRAINTS' section is expanded, showing 'Response Time SLO [ms]' set to 2000, 'Enable Response Time SLO' and 'Enable Transaction SLO' toggled off, and 'Transaction SLO' set to 10.

- Kubernetes cluster

Application owners who do not have access to the Turbonomic user interface can create the following CRs in their Kubernetes clusters. Kuberturno discovers the settings in these CRs and then syncs them with the user interface every ten minutes.

#### – ServicePolicy CR

This is a YAML resource that specifies the policy settings that apply to a group of Kubernetes Services (download a sample [here](#)). The next section discusses how to configure this CR correctly.

The CR should be similar to the following:

```
apiVersion: policy.turbonomic.io/v1alpha1
kind: SLOHorizontalScale
metadata:
 name: Policy_A
spec:
 minReplicas: 1
 maxReplicas: 10
 objectives:
 - name: ResponseTime
 value: 300
 - name: Transaction
 value: 20
 behavior:
 scaleUp: Automatic
 scaleDown: Disabled
```

**Important:**

Before creating the ServicePolicy CR, be sure to add this [CRD](#) to the Kubernetes cluster. This CRD is mandatory and is intended to ensure the validity of the ServicePolicy CR.

**– ServicePolicyBinding CR**

This is a YAML resource that binds the policy settings specified in ServicePolicy to the group of Kubernetes services specified in this CR (download a sample [here](#)). The next section discusses how to configure this CR correctly.

The CR should be similar to the following:

```
apiVersion: policy.turbonomic.io/v1alpha1
kind: PolicyBinding
metadata:
 name: AH-Service_GP
policyRef:
 kind: SLOHorizontalScale
 name: Policy_A
targets:
- apiVersion: apps/v1
 kind: Service
 name: cass-api
- apiVersion: apps/v1
 kind: Service
 name: cass-(cass|user)
```

**Important:**

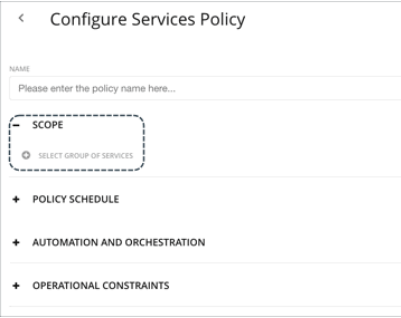
Before creating the ServicePolicyBinding CR, be sure to add this [CRD](#) to the Kubernetes cluster. This CRD is mandatory and is intended to ensure the validity of the ServicePolicyBinding CR.

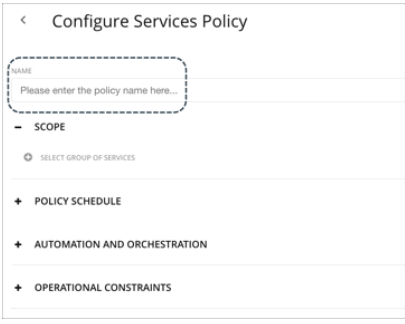
**Settings for Kubernetes Service Policies**

Configure the following settings for the policies.

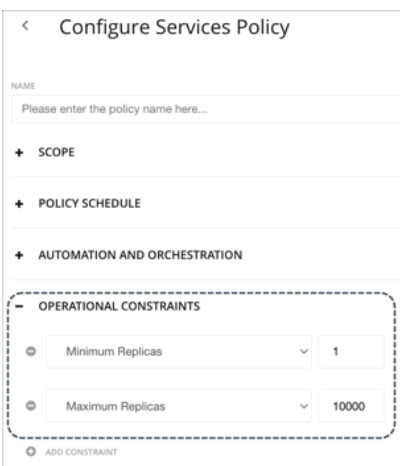
**Note:**

When a group of Services applies multiple conflicting policies, Turbonomic enforces the policy with the most conservative settings.

| Setting | User Interface                                                                                                                                                                                  | Kubernetes Cluster CRs                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Scope   | <p>Specify a group of Kubernetes Services that will apply the policy.</p> <p>Select a group of Services.</p>  | <p>Configure the following in the ServicePolicyBinding CR:</p> <pre> metadata:   name: &lt;Group_Name&gt; policyRef:   kind: SLOHorizontalScale   name: &lt;Policy_Name&gt; targets: - apiVersion: apps/v1   kind: Service   name: &lt;Service_Name_1&gt; - apiVersion: apps/v1   kind: Service   name: &lt;Service_Name_2&gt; </pre> <p>Where:</p> <ul style="list-style-type: none"> <li>• &lt;Group_Name&gt; is the name of the group of Services that will apply the policy. In the user interface, this group will be added to <b>Search &gt; Groups</b> and will be shown in the format &lt;Namespace&gt;/&lt;Group_Name&gt;[Target_Name].</li> <li>• &lt;Policy_Name&gt; is the policy name defined in the ServicePolicy CR. The name appears as follows: <pre> metadata:   name: &lt;Policy_Name&gt; </pre> </li> </ul> <p>In the user interface, this policy will be added to <b>Settings &gt; Policies</b>.</p> <ul style="list-style-type: none"> <li>• Service_Name_1 and Service_Name_2 are the Services belonging to the group. Add more entries as needed. You can specify actual names or use regular expressions.</li> </ul> |

| Setting                    | User Interface                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Kubernetes Cluster CRs                                                                                                                       |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Policy name                | <p>Give the policy a name.</p> <p>Type a name.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | <p>Configure the following in the ServicePolicy CR:</p> <pre>metadata:   name: &lt;Policy_Name&gt;</pre>                                     |
| Horizontal scaling actions | <p>Choose from the following:</p> <ul style="list-style-type: none"> <li>• Turn on scale up and scale down.</li> <li>• Turn on scale up, turn off scale down.</li> <li>• Turn on scale down, turn off scale up.</li> </ul> <p>To turn on, set the action acceptance mode to Automatic, Manual, or Recommend. To turn off, set the action acceptance mode to Disabled. See <a href="#">Action Acceptance Modes</a> for details.</p> <p><b>Note:</b></p> <p>If you turned off horizontal scaling but specified SLOs values in the policy, actions do not generate but SLO values will continue to display in the Response Time and Transaction charts for Services, for your reference. This allows you to gauge performance against those SLOs.</p> | <p>Configure the following in the ServicePolicy CR:</p> <pre>behavior:   scaleUp: &lt;Action_Mode&gt;   scaleDown: &lt;Action_Mode&gt;</pre> |

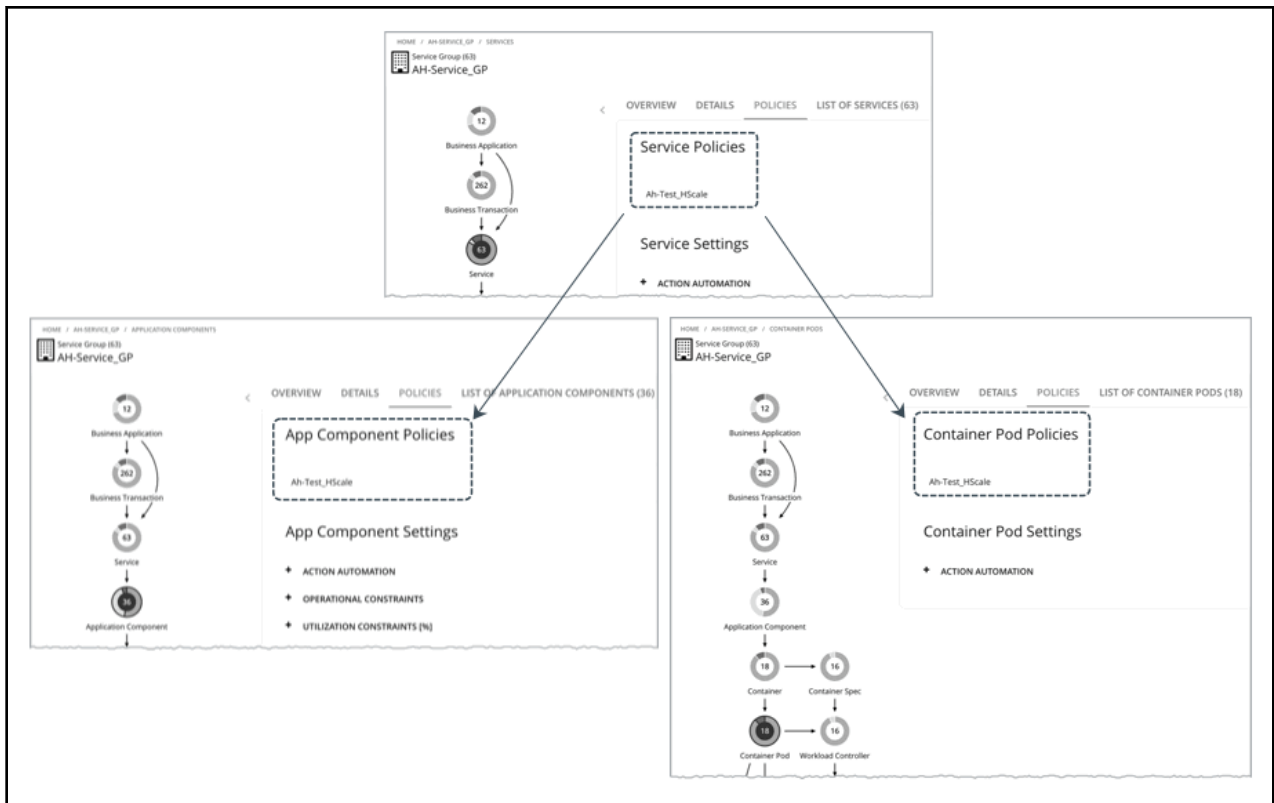


| Setting                                    | User Interface                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Kubernetes Cluster CRs                                                                                                                                                                                                                                                 |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLOs                                       | <p>Set SLOs and your desired SLO values.</p> <ul style="list-style-type: none"> <li>Response Time SLO</li> </ul> <p>Response Time SLO is the desired <i>weighted average</i> response time (in milliseconds) of all Application Component replicas associated with a Service.</p> <ul style="list-style-type: none"> <li>Transaction SLO</li> </ul> <p>Transaction SLO is the maximum number of transactions per second that each Application Component replica can handle.</p> <p>If an SLO value is not specified, the policy will use the default value in the global policy for Services.</p> | <p>Configure the following in the ServicePolicy CR:</p> <pre>objectives: - name: ResponseTime   value: &lt;ResponseTimeSLO&gt; - name: Transaction   value: &lt;TransactionSLO&gt;</pre> <p>To disable a particular SLO, remove the name - value pair from the CR.</p> |
| (Optional)<br>Minimum and Maximum Replicas | <p>Specify your desired values.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <p>Configure the following in the ServicePolicy CR:</p> <pre>spec:   minReplicas: &lt;minReplicas&gt;   maxReplicas: &lt;maxReplicas&gt;</pre>                                                                                                                         |

## Propagation of Service Policy Settings

Settings in a Service policy propagate to the associated pods and Application Components to establish their relationship and provide context.

For example, assume you created a group of Services called AH-Service\_GP and then applied the Service policy Ah-Test\_HScale to that group. When you set the scope to this group, Ah-Test\_HScale displays as a policy in the entity views for Services, Application Components, and Container Pods. You can click the policy name in any view to see or modify the policy settings.



To prevent conflicts, SLO values in Service policies override any SLOs set in Application Components. In addition, the Response Time and Transaction charts for Application Components show SLOs specified in the Service policy.

## Action Automation Considerations

We recommend action automation under the following circumstances:

- Your applications run as a set of Kubernetes services backed by a deployment.
- Services deploy via a namespace *without* quotas.
- Newly provisioned pods are placed on nodes with the same CPU speed.
- You do not have an upstream Kubernetes HPA (Horizontal Pod Autoscaling) enabled for the same workload.

We recommend that you disable automation and manually execute actions under the following circumstances:

- Services deploy via a namespace *with* quotas.
- Newly created pods are scheduled on nodes with different CPU speeds.
- Services have non-resource constraints that could result in newly provisioned pods staying in the pending state.

## Service Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Policies for Non-Kubernetes Services

- **Action Automation and Orchestration**

Turbonomic does not recommend actions for non-Kubernetes Services, but it does recommend actions for the underlying Application Components and nodes. The Pending Actions chart for Services list these actions, thus providing visibility into the risks that have a direct impact on their performance.

- **Transaction SLO**

Enable this SLO if you are monitoring performance through Services.

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                    |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

Transaction SLO determines the upper limit for acceptable transactions per second. When the number of transactions reaches the given value, Turbonomic sets the risk index to 100%.

- **Response Time SLO**

Enable this SLO if you are monitoring performance through Services.

| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                      |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

Response time SLO determines the upper limit for acceptable response time (in milliseconds). If response time reaches the given value, Turbonomic sets the risk index to 100%.

### Policies for Kubernetes Services

These are the policy settings that you configure from the user interface. Turbonomic also supports policy settings configured in CR files in your Kubernetes clusters. For details, see [Actions for Kubernetes Services](#).

- **Action Automation and Orchestration**

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

To generate these actions, you must turn on horizontal scaling (up and/or down) and specify your desired SLOs in a Service policy.

For details about these actions and the environments that are suitable for automating these actions, see [Actions for Kubernetes Services](#).

| Attribute             | Default Setting/Value |
|-----------------------|-----------------------|
| Horizontal Scale Down | Off (Disabled)        |
| Horizontal Scale Up   | Off (Disabled)        |

- **Transaction SLO**

Transaction SLO is the maximum number of transactions per second that each Application Component replica can handle.

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off                                                                                                           |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

- **Response Time SLO**

Response Time SLO is the desired *weighted average* response time (in milliseconds) of all Application Component replicas associated with a Service.

| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off                                                                                                             |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

- **Minimum and Maximum Replicas**

You can adjust the default values based on the characteristics of your applications or if you are planning for capacity. The maximum value also acts as a safeguard against overprovisioning of replicas.

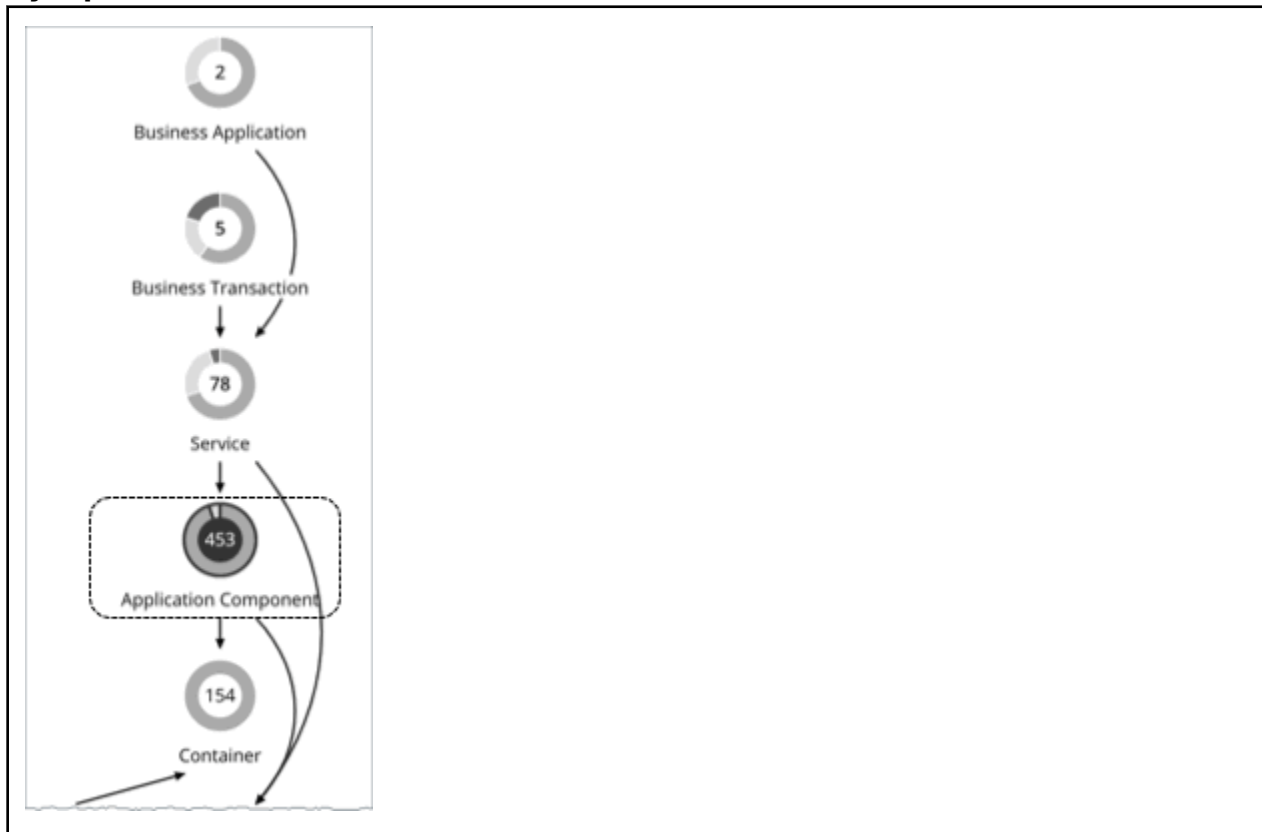
| Attribute        | Default Setting/Value |
|------------------|-----------------------|
| Minimum Replicas | 1                     |
| Maximum Replicas | 10000                 |

## Application Component

An Application Component is a software component, application code, or a unit of processing within a Service that consumes resources to enable it to perform its function for the Business Application. For example, Apache Tomcat is a Java Servlet container that hosts a range of Java applications on the web.

Turbonomic can recommend actions to adjust the amount of resources available to Application Components.

## Synopsis



### Synopsis

|            |                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:    | Application Components have unlimited budget.                                                                                                                                                                                                                                                                                                                                           |
| Provides:  | <ul style="list-style-type: none"> <li>• Response Time and Transactions to Services, <u>Business Transactions</u>, and Business Applications</li> <li>• Response Time, Transactions, Heap, Remaining GC Capacity, and Threads to end users</li> </ul>                                                                                                                                   |
| Consumes:  | Compute resources from nodes                                                                                                                                                                                                                                                                                                                                                            |
| Discovery: | <p>Turbonomic discovers the following:</p> <ul style="list-style-type: none"> <li>• Apache Tomcat</li> <li>• AppDynamics Nodes</li> <li>• Dynatrace Processes</li> <li>• NewRelic APM Application Instances</li> <li>• Application Insights Components</li> <li>• SNMP</li> <li>• WMI</li> <li>• Prometurbio or Data Ingestion Framework metrics for Kubernetes environments</li> </ul> |

## Monitored Resources

The exact resources monitored will differ based on application type. This list includes all resources you may see.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Transactions

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

- Heap

Heap is the portion of a VM or container's memory allocated to individual applications.

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Connection

Connection is the measurement of Database Server connections utilized by applications.

- Remaining GC Capacity

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

- Threads

Threads is the measurement of thread capacity utilized by applications.

The charts for an Application Component show average and peak/low values over time. You can gauge performance against the given SLOs. By default, Turbonomic does not enable SLOs in the default policy for Application Components. It estimates SLOs based on monitored values, but does not use these values in its analysis.

**Note:**

In Kubernetes environments, SLOs defined in a Service policy override any SLOs set in the associated Application Components to prevent conflicts. In addition, the Response Time and Transaction charts for Application Components will show SLOs specified in the Service policy. For more information, see [Actions for Kubernetes Services](#).

## Actions

### Resize

Resize the following resources to maintain performance:

- Thread Pool

Turbonomic generates thread pool resize actions. These actions are recommend-only and can only be executed outside Turbonomic.

- Connections

Turbonomic uses connection data to generate memory resize actions for on-prem Database Servers.

- Heap

Turbonomic generates Heap resize actions if an Application Component provides Heap and Remaining GC Capacity, and the underlying VM or container provides VMem. These actions are recommend-only and can only be executed outside Turbonomic.

**Note:**

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

The resources that Turbonomic can resize depend on the processes that it discovers from your Applications and Databases targets. Refer to the topic for a specific target to see a list of resources that can be resized.

## Application Component Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about Application Component actions, see [Application Component Actions](#).

| Action                          | Default Mode |
|---------------------------------|--------------|
| Resize heap (up or down)        | Recommend    |
| Resize thread pool (up or down) | Recommend    |
| Resize connections (up or down) | Recommend    |

You can use [Action Scripts](#) for action orchestration. Third-party orchestrators (such as ServiceNow) are not supported.

## Transaction SLO

Enable this SLO to monitor the performance of your Application Components.

**Note:**

In Kubernetes environments, SLOs defined in a Service policy override any SLOs set in the associated Application Components to prevent conflicts. In addition, the Response Time and Transaction charts for Application Components will show SLOs specified in the Service policy. For more information, see [Actions for Kubernetes Services](#).

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                    |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

Transaction SLO determines the upper limit for acceptable transactions per second. When the number of transactions reaches the given value, Turbonomic sets the risk index to 100%.

## Response Time SLO

Enable this SLO to monitor the performance of your Application Components.

**Note:**

In Kubernetes environments, SLOs defined in a Service policy override any SLOs set in the associated Application Components to prevent conflicts. In addition, the Response Time and Transaction charts for Application Components will show SLOs specified in the Service policy. For more information, see [Actions for Kubernetes Services](#).

| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                      |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

Response time SLO determines the upper limit for acceptable response time (in milliseconds). If response time reaches the given value, Turbonomic sets the risk index to 100%.

## Heap Utilization

The Heap utilization that you set here specifies the percentage of the existing capacity that Turbonomic will consider to be 100% of capacity. For example, a value of 80 means that Turbonomic considers 80% utilization to be 100% of capacity.

| Attribute            | Default Value |
|----------------------|---------------|
| Heap Utilization (%) | 80            |

Turbonomic uses Heap utilization and Remaining GC Capacity (the percentage of CPU time *not* spent on garbage collection) when making scaling decisions. Assume Heap utilization is at 80%, which is 100% of capacity. However, if Remaining GC Capacity is at least 90% (in other words, CPU time spent on garbage collection is only 10% or less), an 80% Heap utilization does not indicate a shortage after all. As a result, Turbonomic will not recommend Heap scaling.

If Heap utilization is low and Remaining GC Capacity is high, Turbonomic will recommend resizing down Heap. If the opposite is true, then Turbonomic will recommend resizing up Heap.

## Heap Scaling Increment

This increment specifies how many units to add or subtract when scaling Heap for an application component.

| Attribute                   | Default Value |
|-----------------------------|---------------|
| Heap Scaling Increment (MB) | 128           |

Do not set the increment value to be lower than what is necessary for the Application Component to operate. If the increment is too low, then it's possible there would be insufficient Heap for the Application Component to operate. When reducing allocation, Turbonomic will not leave an Application Component with less than the increment value. For example, if you use the default 128, then Turbonomic cannot reduce the Heap to less than 128 MB.

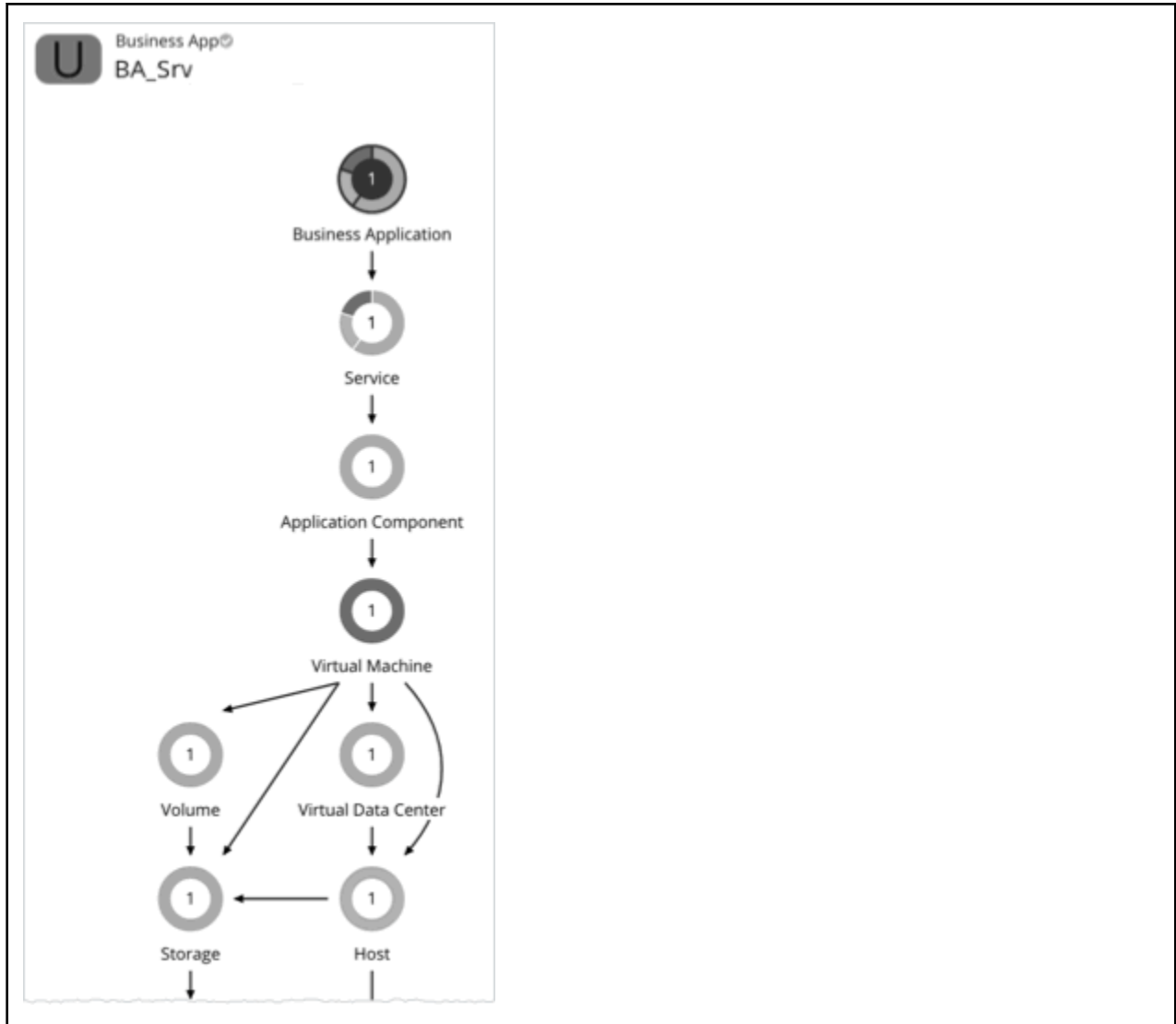
## Application Topology

Turbonomic gives you the ability to create your own [Business Applications](#), [Business Transactions](#), and [Services](#) without the need to ingest additional application data into the platform. This is especially useful



in environments where there are gaps in the application stack shown in the Turbonomic supply chain. For example, in the absence of an application monitoring target such as AppDynamics or Dynatrace, you will not see Business Applications in your supply chain. User-created application entities address those gaps.

When you create a new application entity, you identify interrelated application entities and nodes (i.e., the infrastructure that backs the application entities) in your environment for which you want to measure performance. Turbonomic then links them in a supply chain and represents them as a unified group. You can monitor overall performance for the group in the context of the new application entity, and drill down to the individual entities and nodes for finer details.



Turbonomic does not perform analysis on any user-created application entity, but it aggregates the underlying risks the same way it does for auto-discovered entities.

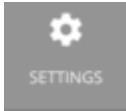
After you create an application entity, Turbonomic counts it in the global supply chain and adds it to the relevant charts (for example, if you created a new Service that has performance risks, you might see it listed in the Top Services chart). Drill down to the newly created entity to monitor its performance. You can also use Search to find the application entity and set it as your scope.

**Note:**

It could take up to 10 minutes to see newly created entities in the supply chain.

## Creating Application Entities

1. Navigate to the **Settings** Page.



2. Choose **Application Topology**.



3. Click **New Application Topology** and then choose *Automatic* or *Manual*.

- Automatic

Create a new application entity composed of tagged entities. For example, create a new Business Application composed of VMs with the "Production" tag.

- a. Select the application entity type that you want to create.
- b. Type an entity name prefix to help you easily identify the application entities that Turbonomic will create for you.
- c. Specify the tags that will identify the underlying entities.

- Manual

Create a new application entity composed of a specific set of application entities and nodes.

- a. Select the application entity type that you want to create.
- b. Give the application entity a name.
- c. Select the underlying application entities and nodes.
- d. Enable or disable **Direct Link**.

- Disabled (default)

When **Direct Link** is disabled, Turbonomic creates a context-based definition of the application entity you are creating and automatically updates that definition as the entity evolves. This allows you to create flexible definitions with minimal effort.

The underlying application entities and nodes that you specified act as "seed entities" for creating the definition. Turbonomic uses these seed entities to identify the highest entity in the supply chain and any other related entities ("leaf entities"), and then creates a new context-based definition. The result is an application topology that closely matches your environment.

For example, your initial intent might be to create a new Business Application entity composed of several Services (seed entities), so you can monitor performance at the Service level.

However, you might not be aware of other entities that could impact performance, making it more time-consuming to identify and resolve performance issues outside of the selected scope. With **Direct Link** disabled, Turbonomic might discover Application Components and VMs (leaf entities) that back the Services, and then show them in the supply chain. The result is a more complete representation of the Business Application that shows performance risks at each level of the discovered application stack. As the composition of the Business Application changes, Turbonomic automatically updates the definition so your supply chain view remains current.

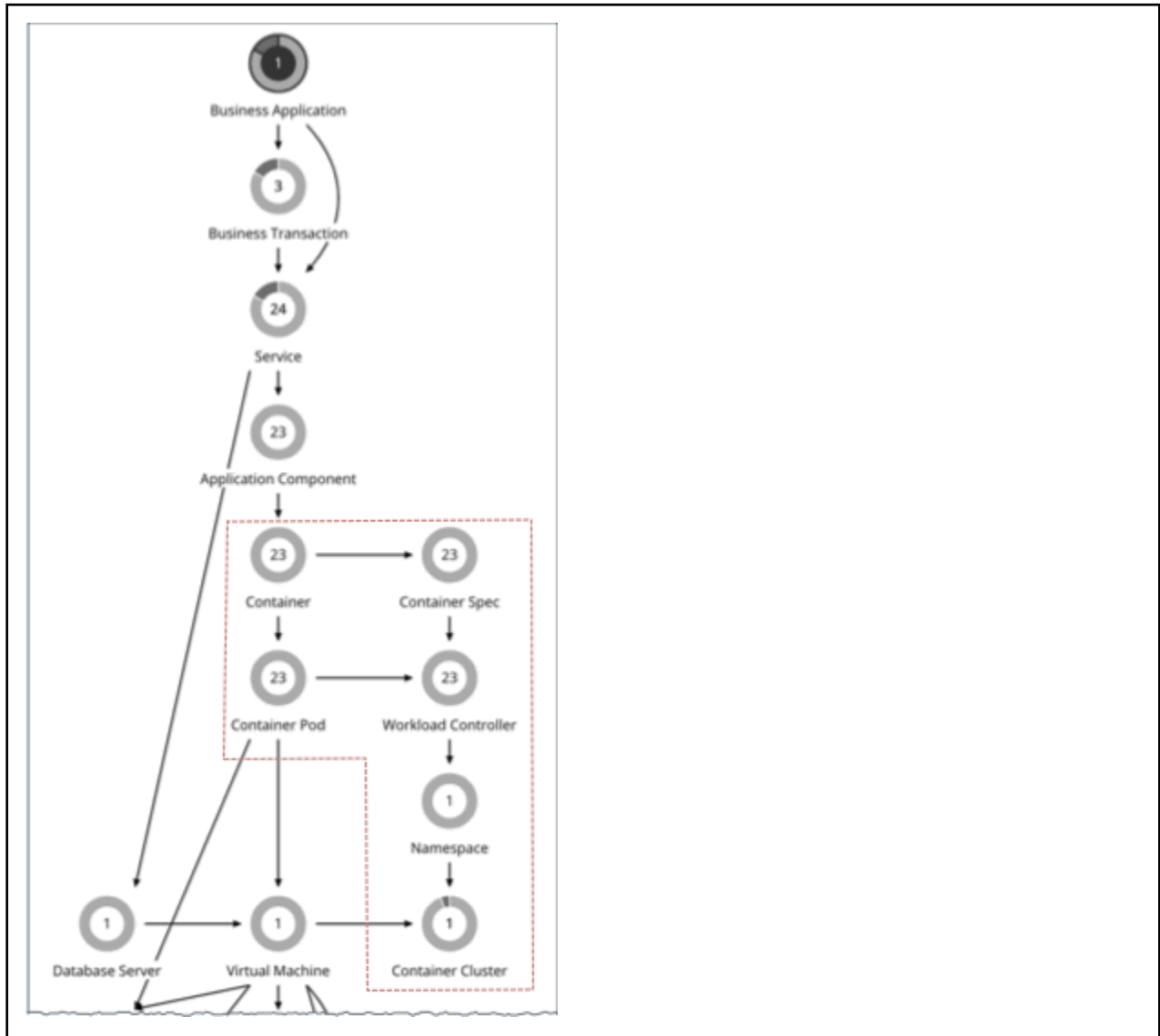
- Enabled

When **Direct Link** is enabled, Turbonomic creates a definition based solely on your selected entities. This option is ideal if you require full control of your definitions. For example, you might have a requirement to limit the scope of your performance monitoring to certain entities.

4. Click **Create Definition**.

## Entity Types - Container Platform

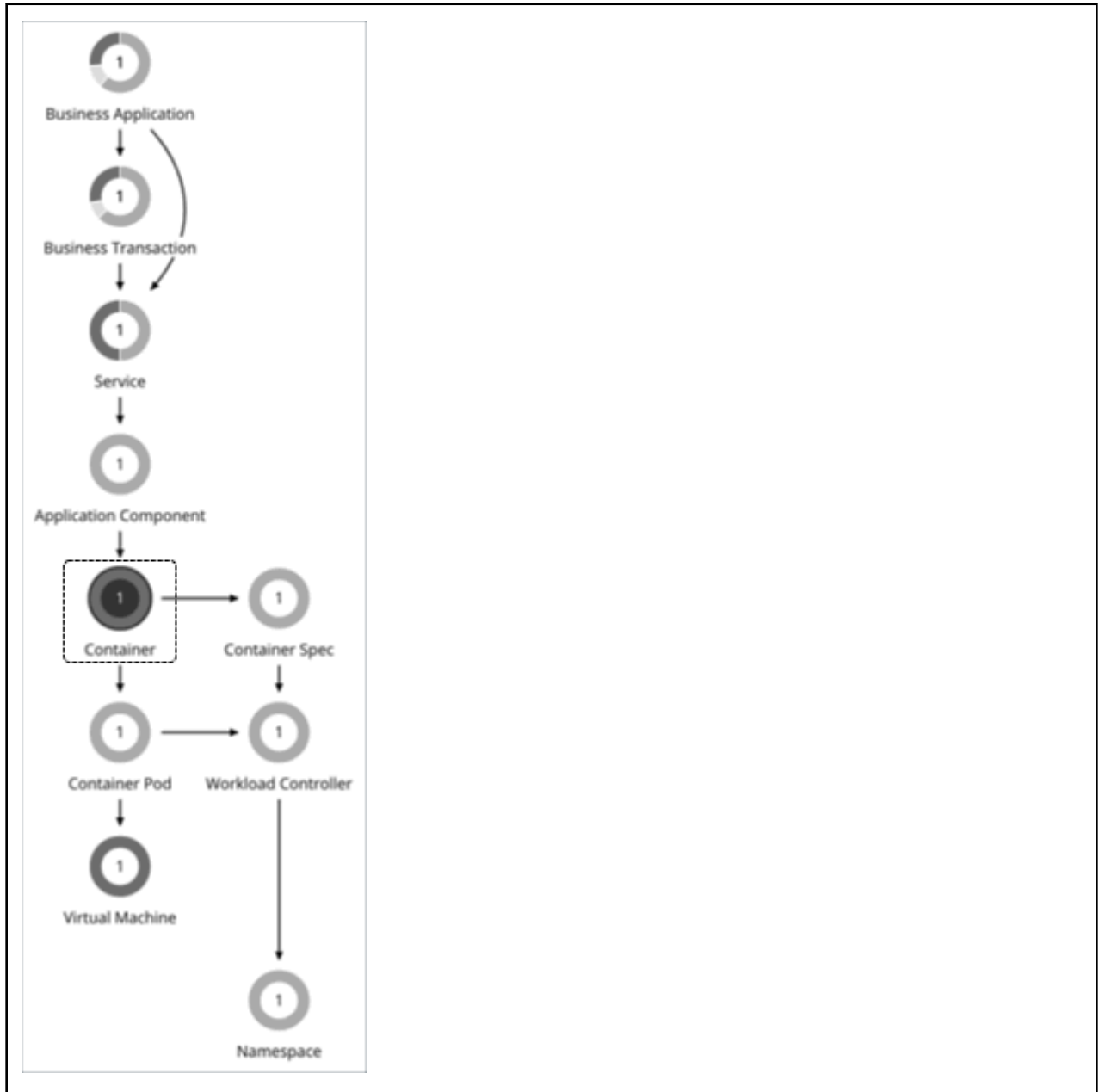
Turbonomic discovers and monitors the entities that make up your container platform, and recommends actions to assure performance for the applications that consume resources from these entities.



### Container

An application container is a standalone, executable image of software that includes components to host an application. Because the container instances that support an application can change at any time, containers are considered *ephemeral*.

## Synopsis



## Synopsis

Provides:

Resources for the applications to use:

- Virtual CPU
- Virtual Memory

Consumes:

Resources from container pods, virtual machines, and virtual datacenters.

## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Discovered through: | <p>For Kubernetes, Turbonomic discovers containers through the Kubeturbo pod that you have deployed in your environment.</p> <p>For Dynatrace and AppDynamics hosted on containers:</p> <ul style="list-style-type: none"><li>• Dynatrace: Turbonomic discovers containers through the metadata of processes.</li><li>• AppDynamics: Turbonomic discovers containers through container objects.</li></ul> |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Monitored Resources

Turbonomic monitors the following resources:

- Virtual Memory (VMem)

VMem is the virtual memory utilized by a container against the memory limit. If no limit is set, node capacity is used.

- VMem Request

If applicable, VMem Request is the virtual memory utilized by a container against the memory request.

- VCPU

VCPU is the virtual CPU (in mCores) utilized by a container against the CPU limit. If no limit is set, node capacity is used).

- VCPU Request

If applicable, VCPU Request is the virtual CPU (in mCores) utilized by a container against the CPU request.

- VCPU Throttling

VCPU Throttling is the throttling of container virtual CPU that could impact response time, expressed as the percentage of throttling for all containers associated with a Container Spec. In the Capacity and Usage chart for containers, *used* and *utilization* values reflect the actual throttling percentage, while *capacity* value is always 100%.

## Actions

### Resize (via Workload Controllers)

Resize containers to assure optimal utilization of resources. By default, containers resize consistently, which allows all replicas of the same container for the same workload type to resize any resource consistently.

Turbonomic shows and executes container resize actions via Workload Controllers. You will *not* see actions when you set the scope to containers.

For details, see [Workload Controller Actions](#).

## Container Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

### Resize (via Workload Controllers)

Resize containers to assure optimal utilization of resources. By default, containers resize consistently, which allows all replicas of the same container for the same workload type to resize any resource consistently.

Turbonomic shows and executes container resize actions via Workload Controllers. You will *not* see actions when you set the scope to containers.

| Action | Default Mode |                      |
|--------|--------------|----------------------|
|        | Container    | Workload Controller  |
| Resize | N/A          | Manual (automatable) |

For details, see [Workload Controller Actions](#).

## Consistent Resizing

- *For groups in user-defined automation policies:*

| Attribute           | Default Setting |
|---------------------|-----------------|
| Consistent Resizing | Off             |

When you create a policy for a group of containers and turn on Consistent Resizing, Turbonomic resizes all the group members to the same size, such that they all support the top utilization of each resource commodity in the group. For example, assume container A shows top utilization of CPU, and container B shows top utilization of memory. Container resize actions would result in all the containers with CPU capacity to satisfy container A, and memory capacity to satisfy container B.

For an affected resize, the Actions List shows individual resize actions for each of the containers in the group. If you automate resizes, Turbonomic executes each resize individually in a way that avoids disruption to your workloads.

- *For auto-discovered groups:*

Turbonomic discovers Kubernetes groups such as Deployments, ReplicationControllers, ReplicaSets, DaemonSets, and StatefulSets, and automatically enables Consistent Resizing in a read-only policy for each group. If you do not need to resize all the members consistently, create another policy for the group and turn off Consistent Resizing.

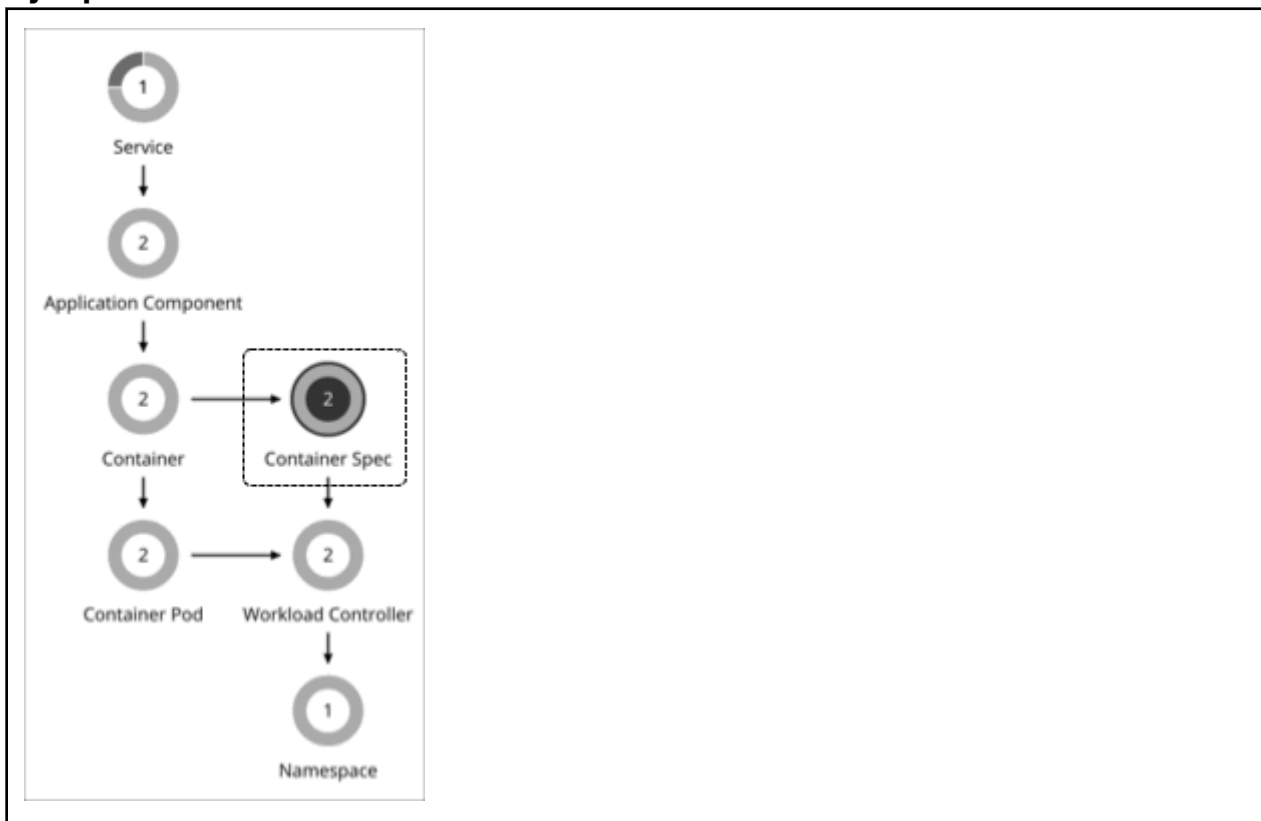
## Container Spec

A Container Spec is a shared definition for all ephemeral container replicas with like properties. It is a persistent entity that maintains historical data for its ephemeral containers, and all the replicas that have run in the past. Turbonomic uses historical data to make container sizing decisions. Data includes:

- vCPU used by all container replicas
- vCPU request capacity (if applicable)
- vMem used by all container replicas
- vMem request capacity (if applicable)

In the Turbonomic supply chain, the count of replicas maps to the count of container entities that a Container Spec encompasses.

## Synopsis



### Synopsis

|                     |                         |
|---------------------|-------------------------|
| Provides:           | N/A                     |
| Consumes:           | N/A                     |
| Discovered through: | Kubeturbo Mediation Pod |

### Monitored Resources

Turbonomic monitors the historical usage of any instance of a container running for the workload (assuming the workload name stays the same). Charts show the trend of usage even with restarts or redeployments.

### Actions

#### None

A Container Spec retains the historical utilization data of ephemeral containers. Turbonomic uses this data to make accurate container resize decisions, but does not recommend actions for the Container Spec itself.

#### Note:

To view container resize actions, set the scope to the Workload Controller for related containers. Go to the Pending Actions chart and click **Show All** to see the full list. For more information about container actions, see [Workload Controller Actions](#).

#### Constraint for Sidecar Container Specs

A Kubernetes service might include [sidecar Container Specs](#) to provide additional services to a running pod, such as security or logging services. Sidecars injected at pod creation cannot be updated from the parent Workload Controller, causing a resize action to fail.

To prevent the execution of resize actions on injected sidecars, Turbonomic adds them to a group called "Injected Sidecars/All ContainerSpecs". This group applies a read-only policy that sets the action acceptance mode for resizes to *Recommend*. This means that you can only execute resizes outside of Turbonomic. The parent Workload Controller will continue to resize non-sidecar Container Specs as usual.

## Container Spec Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

The following settings affect tuned scaling:

| Setting                       | Default Mode |
|-------------------------------|--------------|
| vCPU Request Resize Below Min | Recommend    |
| vCPU Limit Resize Above Max   | Recommend    |
| vCPU Limit Resize Below Min   | Recommend    |
| vMem Request Resize Below Min | Recommend    |
| vMem Limit Resize Above Max   | Recommend    |
| vMem Limit Resize Below Min   | Recommend    |

The default mode of *Recommend* means that when resize values in actions fall outside the normal range (as defined in Container Spec policies), Turbonomic will post the actions for you to review. You can only execute these actions outside Turbonomic. If you set the action mode to *Disabled*, Turbonomic will not generate the actions.

For an overview of tuned resizing, see [Tuned Resizing](#).

## Resize Thresholds

Turbonomic uses resize thresholds as operational constraints to set up tuned scaling for Container Specs. For an overview of tuned scaling, see [Tuned Resizing](#).

| Attribute                                  | Default Value |
|--------------------------------------------|---------------|
| VCPU Request Resize Min Threshold (mCores) | 10            |
| VCPU Limit Resize Min Threshold (mCores)   | 500           |
| VCPU Limit Resize Max Threshold (mCores)   | 64000         |
| VMEM Request Resize Min Threshold (MB)     | 10            |
| VMEM Resize Min Threshold (MB)             | 10            |



| Attribute                      | Default Value |
|--------------------------------|---------------|
| VMEM Resize Max Threshold (MB) | 1048576       |

## Max CPU Throttling Tolerance (%)

This value defines your acceptable level of throttling and directly impacts the resize actions generated on CPU Limits.

| Attribute                        | Default Value |
|----------------------------------|---------------|
| Max CPU Throttling Tolerance (%) | 20            |

A low percentage value indicates more sensitivity to throttling, while a high value indicates more tolerance for throttling and a higher risk of congestion. Turbonomic generates resize actions to accommodate the value that you defined. A default of up to 20% throttling will be tolerated unless defined otherwise.

Learn more about CPU throttling [here](#).

## Increment Constants

Turbonomic recommends changes in terms of the specified resize increments.

| Attribute                                                   | Default Value |
|-------------------------------------------------------------|---------------|
| Increment constant for VCPU Limit and VCPU Request (mCores) | 100           |
| Increment constant for VMEM Limit and VMEM Request (MB)     | 128           |

For example, assume the vCPU request increment is 100 mCores and you have requested 800 mCores for a container. Turbonomic could recommend to reduce the request by 100, down to 700 mCores.

For vMem, you should not set the increment value to be lower than what is necessary for the container to operate. If the vMem increment is too low, then it's possible that Turbonomic would allocate insufficient vMem. For a container that is underutilized, Turbonomic will reduce vMem allocation by the increment amount, but it will not leave a container with zero vMem. For example, if you set this to 128, then Turbonomic cannot reduce the vMem to less than 128 MB.

## Rate of Resize

(For the *default* policy only)

| Attribute      | Default Value |
|----------------|---------------|
| Rate of Resize | High          |

When resizing resources for a container, Turbonomic calculates the optimal values for vMem and vCPU. But it does not necessarily make a change to that value in one action. Turbonomic uses the Rate of Resize setting to determine how to make the change in a single action, as follows:

- **Low**

Change the value by one increment, only. For example, if the resize action calls for increasing vMem, and the increment is set at 128, Turbonomic increases vMem by 128 MB.

- **Medium**

Change the value by an increment that is 1/4 of the difference between the current value and the optimal value. For example, if the current vMem is 2 GB and the optimal vMem is 10 GB, then Turbonomic will raise vMem to 4 GB (or as close to that as the increment constant will allow).

- **High**

Change the value to be the optimal value. For example, if the current vMem is 2 GB and the optimal vMem is 8 GB, then Turbonomic will raise vMem to 8 GB (or as close to that as the increment constant will allow).

## Aggressiveness and Observation Periods

Turbonomic uses these settings to calculate utilization percentiles for vCPU and vMEM. It then recommends actions to improve utilization based on the observed values for a given time period.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 99th Percentile |

When evaluating vCPU and vMEM performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 99th percentile. The percentile utilization is the highest value that 99% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce the capacity for CPU on a container. Without using a percentile, Turbonomic never resizes below the recognized peak utilization. For most containers there are moments when peak CPU reaches high levels. Assume utilization for a container peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce allocated CPU for that container.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single CPU burst to 100%, but for 99% of the samples CPU never exceeded 50%. If you set **Aggressiveness** to 99th Percentile, then Turbonomic can see this as an opportunity to reduce CPU allocation for the container.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 100th Percentile – The least aggressive, recommended for critical workloads that need maximum guaranteed performance at all times.
- 99th Percentile (Default) – The recommended setting to achieve maximum performance and savings.
- 90th Percentile – Most aggressive, recommended for non-production workloads that can stand higher resource utilization.

- **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 30 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. (If the database has fewer days' data then it uses all of the stored historical data.)

A shorter period means there are fewer data points to account for when Turbonomic calculates utilization percentiles. This results in more dynamic, elastic resizing, while a longer period results in more stable or less elastic resizing. You can make the following settings:

- Less Elastic – Last 90 Days
- Recommended – Last 30 Days
- More Elastic – Last 7 Days

- **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | 1 Day         |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

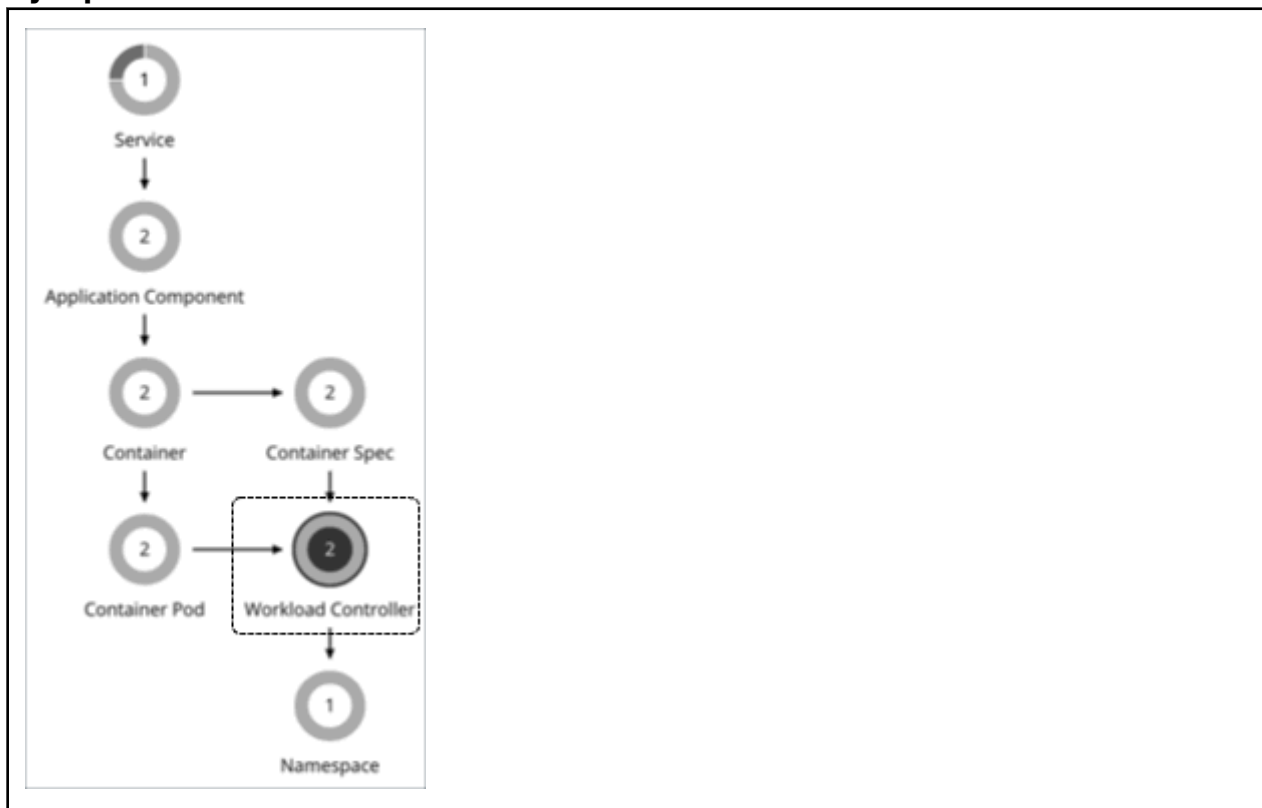
- More Elastic – None
- Recommended – 1 Day
- Less Elastic – 3 or 7 Days

## Workload Controller

A Workload Controller is a Kubernetes controller that watches the state of your pods and then requests changes where needed. Examples of Kubernetes controllers are Deployments and StatefulSets. A single Workload Controller can contain one or more Container Spec entities, and it can be related to one or more running replica pods. Like Container Specs, Workload Controllers are persistent.

You can execute container resize actions when you set the scope to a Workload Controller.

## Synopsis



## Synopsis

|                     |                         |
|---------------------|-------------------------|
| Budget:             | N/A                     |
| Provides:           | N/A                     |
| Consumes:           | N/A                     |
| Discovered through: | Kubeturbo Mediation Pod |

## Monitored Resources

Turbonomic monitors quotas (limits and requests) for VCPU and VMem.

## Workload Controller Actions

### Resize Containers

A Workload Controller executes container actions. When you set the scope to a Workload Controller and view the actions list, the actions apply to containers. Turbonomic does not recommend actions for the Workload Controller itself.

For vCPU limit resizes, Turbonomic will recommend a resize up action, even if utilization percentile is low, to address slow response times associated with CPU throttling.

CPU throttling occurs when you configure a CPU limit on a container, which can inadvertently slow your applications' response time. Even if you have more than enough resources on your underlying node, your container workload will still be throttled because it was not configured properly. High response times are directly correlated to periods of high CPU throttling, and this is exactly how Kubernetes was designed to work. Learn more about CPU throttling [here](#).

Especially for sudden throttling spikes, Turbonomic will persist the related resize actions so you can evaluate these actions even after the spikes have gone away, and then execute them to prevent spikes from re-occurring. As throttling drops, Turbonomic will not recommend a resize down action right away, as this could result in subsequent back-and-forth upsize and downsize recommendations. Instead, it evaluates past throttling to decide when a resize down action is finally safe to execute. To ensure the timeliness of these actions and arrive at the optimal resize values to recommend, Turbonomic calculates fast and slow moving throttling averages, and then displays *smoothed* and *daily* averages in charts.

Smoothed average is an exponential moving average and moving variance method used on CPU throttling data. It allows analysis to generate a vCPU limit resize up action more quickly when throttling is detected, and be conservative on resize down to mitigate introducing throttling.

## Action Propagation

Actions to resize containers via Workload Controllers propagate to application entities and the underlying container infrastructure to show the impact of these actions on the health of your applications and container environment.

## Action Merging

Executing several container resize actions via Workload Controllers can be very disruptive since pods need to restart with each resize. For replicas of the container scale group(s) related to a single Workload Controller, Turbonomic consolidates resize actions into one *merged action* to minimize disruptions. When a merged action has been executed (via the associated Workload Controller), all resizes for all related container specifications will be changed at the same time, and pods will restart once.

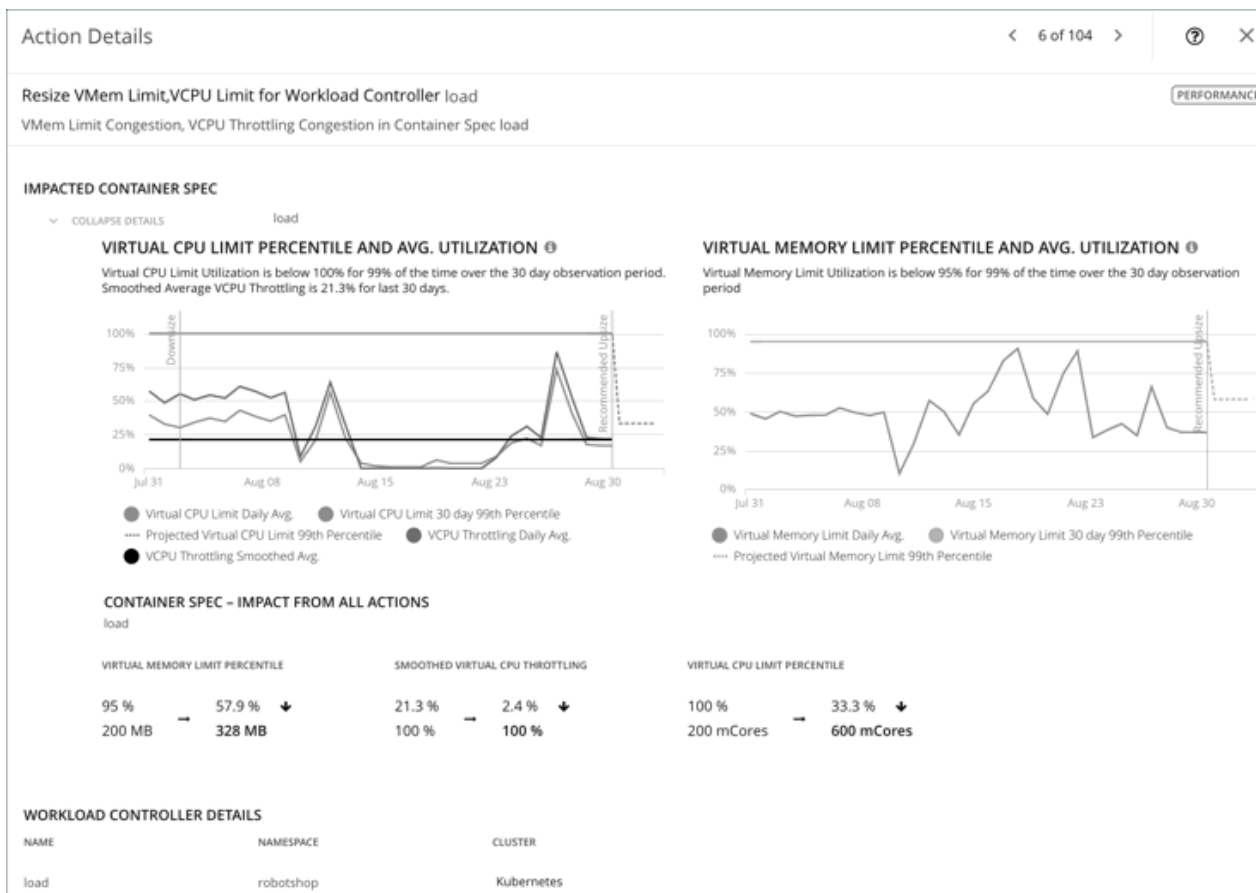
## Action Execution

After you set the scope to Workload Controllers, go to the Pending Actions chart and then click **Show All** to see the full list of resize actions that you can execute. This list includes individual and merged actions. You can filter the list to focus on specific actions, such as actions to address resource congestion or vCPU throttling.

| Action Description                                                                             | Risk                                           | Action Category | Action  |
|------------------------------------------------------------------------------------------------|------------------------------------------------|-----------------|---------|
| Resize VCPU Limit for Workload Controller (f-default-backend)                                  | VCPU Throttling Congestion in Container Spe... | PERFORMANCE     | DETAILS |
| Resize VCPU Limit for Workload Controller add-on-http-application-routing-default-http-backend | VCPU Throttling Congestion in Container Spe... | PERFORMANCE     | DETAILS |
| Resize VMem Limit, VCPU Limit for Workload Controller mypod                                    | Underutilized VMem Limit, VCPU Throttling C... | PERFORMANCE     | DETAILS |
| Resize VCPU Request, VMem Limit, VCPU Limit for Workload Controller twitter-case-friend        | Underutilized VCPU Request, Underutilized V... | PERFORMANCE     | DETAILS |
| Resize VMem Limit, VCPU Limit for Workload Controller rabbitmq                                 | Underutilized VMem Limit, VCPU Throttling C... | PERFORMANCE     | DETAILS |
| Resize VMem Limit, VCPU Limit for Workload Controller load                                     | VMem Limit Congestion, VCPU Throttling Con...  | PERFORMANCE     | DETAILS |

By default, container resize actions are set in *Manual* mode at the Workload Controller level. This means that Turbonomic will not execute any action automatically, and you can manually select the actions that you want to execute. If you prefer to execute actions outside Turbonomic, create Workload Controller policies and set the resize action acceptance mode to *Recommend*. To automate actions, create Workload Controller policies and set the resize action acceptance mode to *Automatic*.

For each action, click DETAILS and expand the Details section to view time series charts that explain the reason for the action. These charts highlight *utilization percentiles* and *smoothed throttling averages* for a given observation period. Turbonomic uses percentile calculations to make accurate resize decisions.



These charts also:

- Plot daily average percentiles and throttling, for your reference.
- Show projected percentiles after you execute the action. If you have previously executed resize actions on the same Workload Controller, the charts show the resulting improvements in daily average utilization.

Put together, these charts allow you to easily recognize trends that drive Turbonomic's resize recommendations.

**Note:**

You can set scaling constraints in Container Spec policies to refine the percentile calculations. For details, see [Aggressiveness and Observation Periods](#).

### Blocking of Action Execution Due to Insufficient Quotas

Turbonomic treats quotas defined in a namespace as constraints when making container resize decisions. If existing container actions would exceed the namespace quotas, Turbonomic recommends actions to resize up the affected namespace quota.

**Note:**

For more information about namespace quotas, see [Namespace Monitored Resources](#).

The action details for a Workload Controller include descriptions of the affected Container Spec entities, and how the resources will change for each. If the resize exceeds current namespace quotas, then Turbonomic blocks the Workload Controller action. The action details list the Namespace actions that block execution of this resize in the **Related Actions** list.

## Action Details

### Resize VCPU Limit, VMem Limit for Workload Controller `cpu-quota-3`

VCPU Throttling Congestion, VMem Limit Congestion in Container Spec `cpu-quota-3-spec`

#### IMPACTED CONTAINER SPEC

COLLAPSE DETAILS

`cpu-quota-3-spec`



STATE

Action execution is blocked by related actions.

#### RELATED ACTIONS

BLOCKED BY

Resize up VCPU Limit Quota for Namespace `quota-test-with-down` from 3,200 mCores to 3,600 mCores in EA - Advanced Engineering

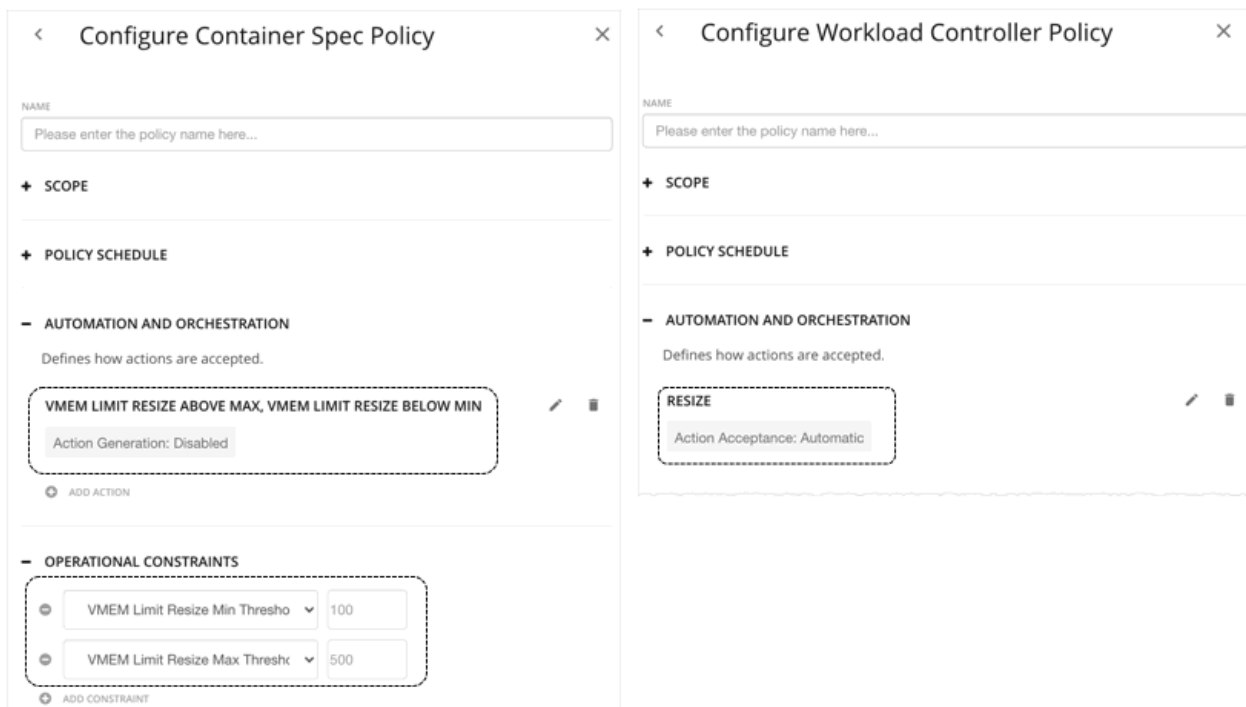
Resize up VMem Limit Quota for Namespace `quota-test-with-down` from 3.4 GB to 4 GB in EA - Advanced Engineering

## Tuned Resizing

Turbonomic can automate container resizes if the resize values fall within a normal range, and then post more conservative actions when resize values fall outside the range. To do this, you would set specific *action acceptance modes* and *tuned resizing* values in policies.

For example, consider resizing vMem limits. As memory demand increases, Turbonomic can automatically execute vMem limit resizes that fall within the normal range. If the Container Spec requests memory beyond the normal range, Turbonomic will either ignore the action or post it for you to review, depending on the tuned resizing settings that you configured.

Assume the following tuned scaling settings in policies:



- The **Operational Constraints** settings in the Container Spec policy specify 100 MB to 500 MB as the normal range.
- With the **Resize** action acceptance mode in the Workload Controller policy set to *Automatic*, Turbonomic will automate *resize up* actions that are below the *max* threshold and *resize down* actions that are above the *min* threshold.

**Note:**

If the action acceptance mode is *Recommend*, Turbonomic will post the actions for you to review. You can only execute these actions outside Turbonomic.

- In the Container Spec policy, since the action acceptance mode for **vMem Limit Resize Above Max** and **vMem Limit Resize Below Min** is set to *Disabled*, Turbonomic will not generate resize actions that fall outside the normal range.
- Since vMEM increment constant is *not* defined in the Container Spec policy, Turbonomic uses the default value of 128 MB.

With these two policies in effect:

- If a Container Spec with 200 MB of vMEM limit needs to resize to 328 MB, Turbonomic automatically resizes to 328 MB.
- If a Container Spec with 200 MB of vMEM limit needs to resize to 72 MB, Turbonomic does not generate the action. vMEM limit remains at 200 MB.

**Note:**

Action policies include scope to determine which entities will be affected by the given policy. It's possible for two or more policies to affect the same entities. As is true for other policy settings, Turbonomic uses the most conservative settings for the affected entities.

For tuned resizing, the effective action acceptance mode will be the most conservative, and the effective tuned resizing range will be the narrowest range (the lowest *Max* and highest *Min*) out of the multiple policies that affect the given entities. For more information, see [Default and User-defined Policies](#).



## Downloading Data for Executed Resize Actions

You can view and download data for executed resize actions to gain insight into workloads with performance risks due to resource limits, as well as opportunities to reclaim unused resource requests.

1. Add the Actions chart to your dashboard.
2. Configure the chart. Be sure to set the scope to Workload Controllers to narrow the results, select Tabular as the chart type, and then set the filter to either All Actions or Executed Actions.
3. In the chart, click **Show All**.
4. In the page that opens, click the download button at the top-right section of the page. If you set All Actions as the filter, select **Executed Actions**.
5. Open the downloaded file and go to the Resize Workload Controller sheet.

Pay attention to the following columns:

- Namespace, Container Spec, and Container Cluster columns – Data in these columns is the most up-to-date at the time of download.
- Change column – This column highlights the impact of resize actions on the resources allocated to containers. Each value represents the difference between the Current Value and New Value columns.
- Container Spec column – If there is more than one Container Spec in a Workload Controller, you can sort this column to easily identify resize actions on individual Container Specs.
- Impacted Commodity column – This column shows if VMem or VCPU was resized. A resize action might show both commodities, which indicates that it is a *merged* action (an action that consolidates resizes related to a single Workload Controller, to minimize disruptions).

## Workload Controller Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

A Workload Controller executes [container](#) actions. When you set the scope to a Workload Controller and view the actions list, the actions apply to containers. Turbonomic does not recommend actions for the Workload Controller itself.

| Action | Default Mode |                      |
|--------|--------------|----------------------|
|        | Container    | Workload Controller  |
| Resize | N/A          | Manual (automatable) |

For details, see [Workload Controller Actions](#).

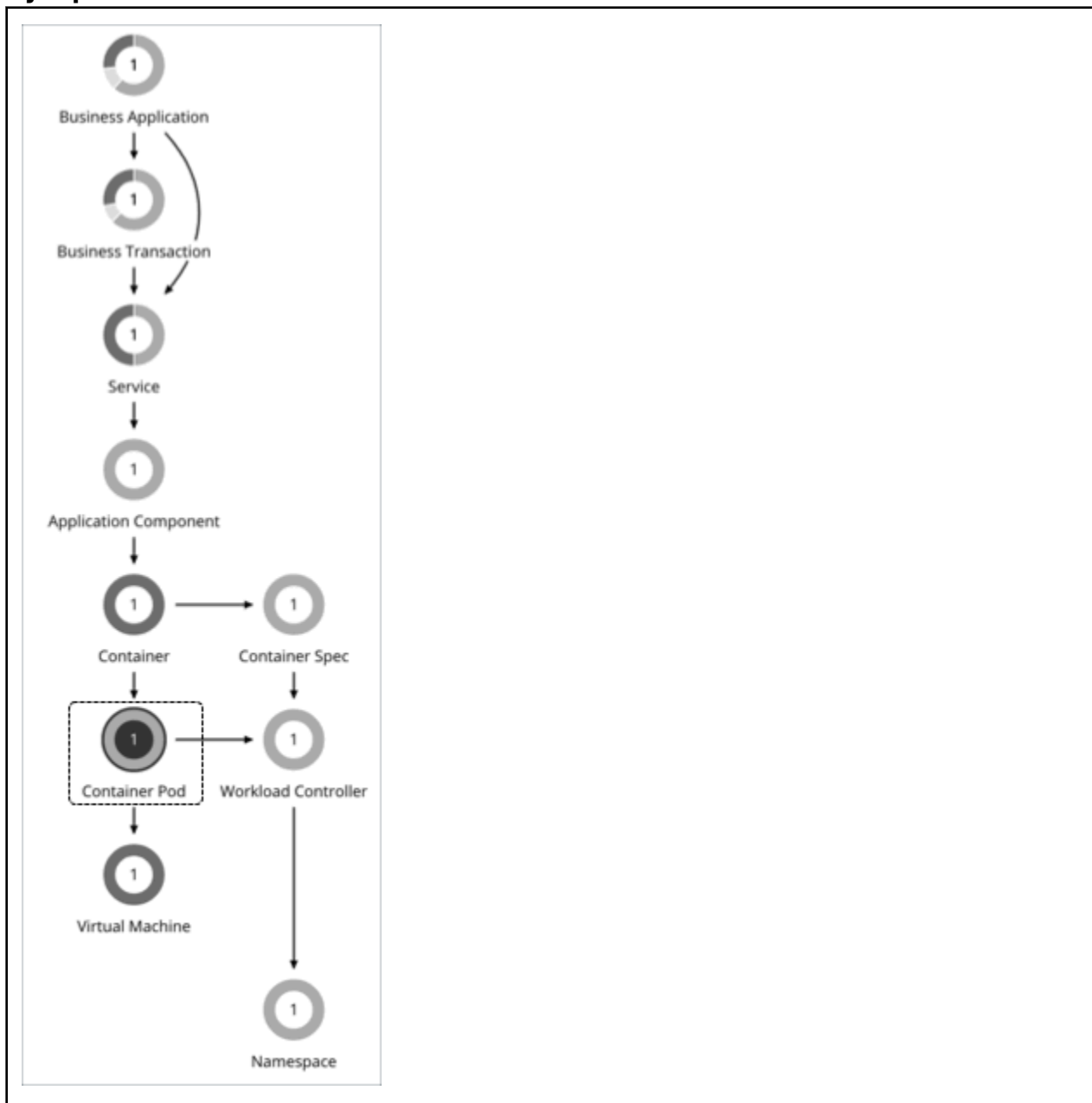
Action orchestration is currently not supported.

## Container Pod

A container pod is a group of one or more containers with shared storage or network resources. It is the smallest deployable unit of computing that you can create and manage in Kubernetes.

A container pod specifies how to run containers together. Like containers, a container pod is *ephemeral*.

## Synopsis



### Synopsis

|                     |                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------|
| Provides:           | Resources for containers to use: <ul style="list-style-type: none"> <li>• Virtual CPU</li> <li>• Virtual Memory</li> </ul> |
| Consumes:           | Resources from virtual machines and namespaces.                                                                            |
| Discovered through: | Turbonomic discovers container pods through the Kubeturbo pod that you have deployed in your environment.                  |

### Monitored Resources

Turbonomic monitors the following resources:

- VMem  
VMem is the virtual memory utilized by a pod against the node physical capacity.
- VMem Request  
VMem Request is the virtual memory request allocated by a pod against the node allocatable capacity.
- VCPU  
VCPU is the virtual CPU (in mCores) utilized by a pod against the node physical capacity.
- VCPU Request  
VCPU Request is the virtual CPU request (in mCores) allocated by a pod against the node allocatable capacity.
- VMem Request Quota  
If applicable, VMem Request Quota is the amount of virtual memory request a pod has allocated against the namespace quota.
- VCPU Request Quota  
If applicable, VCPU Request Quota is the amount of virtual CPU request (in mCores) a pod has allocated against the namespace quota.
- VMem Limit Quota  
If applicable, VMem Limit Quota is the amount of virtual memory limit a pod has allocated against the namespace quota.
- VCPU Limit Quota  
If applicable, VCPU Limit Quota is the amount of virtual CPU limit (in mCores) a pod has allocated against the namespace quota.

## Actions

Turbonomic can recommend the following actions:

- **Move**

Move a pod between nodes (VMs) to address performance issues or improve infrastructure efficiency. For example, if a particular node is congested for CPU, you can move pods to a node with sufficient capacity. If a node is underutilized and is a candidate for suspension, you must first move the pods before you can safely suspend the node.

- **Provision/Suspend**

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, provision or suspend pods associated with those Services to maintain SLOs for your applications.

When recommending node provision or suspend actions, Turbonomic will also recommend provisioning pods (based on demand from DaemonSets) or suspending the related pods.

## Pod Move Actions

The following items impact the generation and execution of pod move actions:

- Constraints

Turbonomic respects the following constraints when making placement decisions for pods:

- Kubernetes taints for nodes and tolerations for pods are treated as constraints. For example, if a pod has a toleration attribute that restricts it from moving to a certain node, Turbonomic will not move that pod to the restricted node.
- Turbonomic imports Kubernetes node labels and treats them as constraints. For example, if a pod has a defined node label, Turbonomic will move that pod to a node with a matching label.

- Turbonomic recognizes pod affinity and anti-affinity policies.
- You can create placement policies to enforce constraints for pod move actions. For example, you can have a policy that allows pods to only move to certain nodes, or a policy that prevents pods from moving to certain nodes.

For more information, see [Creating Placement Policies](#).

- Eviction Thresholds

Turbonomic considers the memory/storage eviction thresholds of the destination node to ensure that the pod can be scheduled after it moves. Eviction thresholds for `imagefs` and `rootfs` are reflected as node effective capacity in the market analysis.

- Temporary Quota Increases

If a namespace quota is already fully utilized, Turbonomic temporarily increases the quota to allow a pod to move, while maintaining that one replica continues to run.

- Security Context Constraints (SCCs)

Red Hat OpenShift uses [SCCs](#) to control permissions for pods. This translates to permissions that users see within the containers of the pods, and the permissions for the processes running inside those pods.

When executing pod move actions, Kubeturbo normally runs with OpenShift cluster administrator permissions to create a new pod and remove the old one. Because of this, the SCCs for the new pod are those that are available to a cluster administrator. It is therefore possible for the new pod to run with an SCC that has higher privileges than the old pod. For example, an old pod might have `restricted_scc` access, while the new one might have `anyuid_scc` access. This introduces a privilege escalation issue.

To prevent privilege escalation when moving pods, Kubeturbo enforces user impersonation, which carries the user-level SCCs of the old pod over to the new pod. To enforce user impersonation, Kubeturbo performs the following tasks:

- Create a [user impersonation account](#) for each SCC level.
- Create a [service account](#) and treat it as a user account for each SCC level currently running in a given cluster.
- Provide [role-based access](#) to SCCs used for impersonation via the service accounts. A service account is allowed to use only one SCC resource in the cluster.
- Create a `role` binding resource to allow service account access to a particular role.

All resources created to enforce user impersonation are removed when Kubeturbo shuts down.

Be aware that by default, an arbitrary pod running in a given cluster does not recognize the namespace it is configured to run in, which is a requirement for user impersonation enforcement. For Kubeturbo to recognize the namespaces for pods, we recommend adding an environment variable named `KUBETURBO_NAMESPACE` via the [downward API](#). Our standard installation methods add the following environment variable to the Kubeturbo deployment spec.

```
env:
 - name: KUBETURBO_NAMESPACE
 valueFrom:
 fieldRef:
 fieldPath: metadata.namespace
```

With this environment variable, Kubeturbo can successfully create the resources needed to enforce user impersonation. Without this variable, Kubeturbo creates the resources in the namespace called `default`. This might cause issues if you need to run multiple instances of Kubeturbo in the same cluster. For example, one instance might run as an observer, and another as an administrator. To ensure multiple Kubeturbo instances within the same cluster do not conflict when creating and removing user impersonation resources, run the instances in separate namespaces.

## Pod Provision and Suspension Actions In Response to SLOs

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

For details, see [Actions for Kubernetes Services](#).

## Pod Provision Action in Response to Node Provision

When recommending node provision actions, Turbonomic also recommends pod provision actions that reflect the projected demand from required DaemonSet pods, and respects the maximum number of pods allowed for a node. This ensures that any application workload can be placed on the new node and stay within the desired range of vMem/vCPU usage, vMem/vCPU request, and number of consumers.

The action details for a pod provision action shows the related node that you need to provision. Click the node name to set it at your scope.

The screenshot shows the 'Action Details' page for a pod provision action. The page is titled 'Action Details' and has a navigation bar with '< 1 of 25 >' and a help icon. The main content area is divided into several sections:

- Provision Container Pod similar to kube-system/nodelocaldns-2pcw6**: This section is highlighted with a dashed box. It contains the text 'Clone kube-system/nodelocaldns-2pcw6 on cloned node1' and a 'PERFORMANCE' button.
- CONTAINER POD DETAILS**: This section contains a table with the following columns: NAME, WORKLOAD CONTROLLER, NAMESPACE, and CONTAINER CLUSTER. The table has one row with the following values: nodelocaldns-2p..., nodelocaldns, kube-system, and Kubernetes-Turb....
- TAGS**: This section contains three tags: controller-revision-hash: 599484f714, pod-template-generation: 1, and k8s-app: nodelocaldns.
- CURRENT CONTAINER POD - IMPACT FROM ALL ACTIONS**: This section contains the text 'kube-system/nodelocaldns-2pcw6' and 'No Impact'.
- STATE**: This section contains the text 'Action acceptance is blocked by policy or system. Acceptance mode is Recommend.'
- RELATED ACTIONS**: This section is highlighted with a dashed box and contains the text 'CAUSED BY' and 'Provision Virtual Machine similar to node1'.

Turbonomic treats static pods as DaemonSets for the purpose of provisioning nodes. Because a static pod provides a node with a specific capability, it is controlled by the node and is not accessible through the API server. If a node to be provisioned requires a static pod, Turbonomic generates actions to provision the node and the corresponding static pod.

Turbonomic creates an auto-generated group of static pods when it discovers a static pod on each node in a cluster. To view all the auto-generated groups, go to Search, select Groups, and then type `mirror pods` as your search keyword.

Search  
Search within your infrastructure

Accounts  
App Component Specs  
Application Components  
Billing Families  
Business Applications  
Business Transactions  
Business Users  
Chassis  
Clusters  
Container Platform Clusters  
Container Pods  
Container Specs  
Containers  
Data Centers  
Database Servers  
Databases  
Desktop Pools  
Disk Arrays  
Folders  
Groups  
Hosts

mirror pods

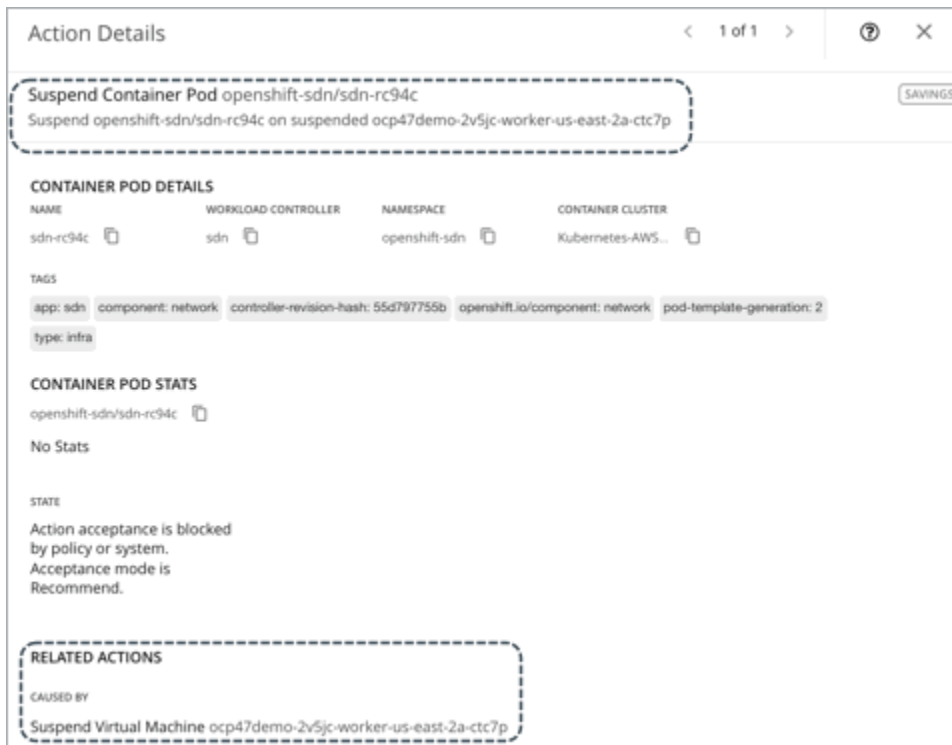
ADD FILTER

|                                                  |                               |        | NAME ↑ |
|--------------------------------------------------|-------------------------------|--------|--------|
| Mirror Pods Kubernetes-ae-cluster-1              | 5                             | Static | >      |
| <small>On-Prem   Kubernetes-ae-cluster-1</small> | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-ae-cluster-2              | 5                             | Static | >      |
| <small>On-Prem   Kubernetes-ae-cluster-2</small> | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-DC11-PT-KBs               | 12                            | Static | >      |
| <small>On-Prem   Kubernetes-DC11-PT-KBs</small>  | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-Hybrid                    | 4                             | Static | >      |
| <small>Hybrid   Kubernetes-Hybrid</small>        | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-OCP43-AWS                 | 8                             | Static | >      |
| <small>Cloud   Kubernetes-OCP43-AWS</small>      | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-OKD-311                   | 9                             | Static | >      |
| <small>On-Prem   Kubernetes-OKD-311</small>      | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-Turbonomic                | 3                             | Static | >      |
| <small>Hybrid   Kubernetes-Turbonomic</small>    | <small>Container Pods</small> |        |        |

## Pod Suspension Action in Response to Node Suspension

When recommending node suspension actions, Turbonomic also recommends suspending the DaemonSet pods that are no longer required to run the suspended nodes.

The action details for a pod suspension action shows the related node that you need to suspend. Click the node name to set it at your scope.



Turbonomic treats static pods as DaemonSets for the purpose of suspending nodes. Because a static pod provides a node with a specific capability, it is controlled by the node and is not accessible through the API server. If the only workload type left on a node is a static pod, Turbonomic generates actions to suspend the node and the corresponding static pod.

Turbonomic creates an auto-generated group of static pods when it discovers a static pod on each node in a cluster. To view all the auto-generated groups, go to Search, select Groups, and then type `mirror pods` as your search keyword.

Search  
Search within your infrastructure

Accounts

App Component Specs

Application Components

Billing Families

Business Applications

Business Transactions

Business Users

Chassis

Clusters

Container Platform Clusters

Container Pods

Container Specs

Containers

Data Centers

Database Servers

Databases

Desktop Pools

Disk Arrays

Folders

Groups

Hosts

mirror pods

ADD FILTER

|                                                  |                               |        | NAME ↑ |
|--------------------------------------------------|-------------------------------|--------|--------|
| Mirror Pods Kubernetes-ae-cluster-1              | 5                             | Static | >      |
| <small>On-Prem   Kubernetes-ae-cluster-1</small> | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-ae-cluster-2              | 5                             | Static | >      |
| <small>On-Prem   Kubernetes-ae-cluster-2</small> | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-DC11-PT-KBs               | 12                            | Static | >      |
| <small>On-Prem   Kubernetes-DC11-PT-KBs</small>  | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-Hybrid                    | 4                             | Static | >      |
| <small>Hybrid   Kubernetes-Hybrid</small>        | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-OCP43-AWS                 | 8                             | Static | >      |
| <small>Cloud   Kubernetes-OCP43-AWS</small>      | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-OKD-311                   | 9                             | Static | >      |
| <small>On-Prem   Kubernetes-OKD-311</small>      | <small>Container Pods</small> |        |        |
| Mirror Pods Kubernetes-Turbonomic                | 3                             | Static | >      |
| <small>Hybrid   Kubernetes-Turbonomic</small>    | <small>Container Pods</small> |        |        |

## Container Pod Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about container pod actions, see [Container Pod Actions](#).

| Action | Default Mode |
|--------|--------------|
| Move   | Manual       |

Action orchestration is currently not supported.

## Placement Policies

You can create placement policies to enforce constraints for pod move actions. For example, you can have a policy that allows pods to only move to certain nodes, or a policy that prevents pods from moving to certain nodes.

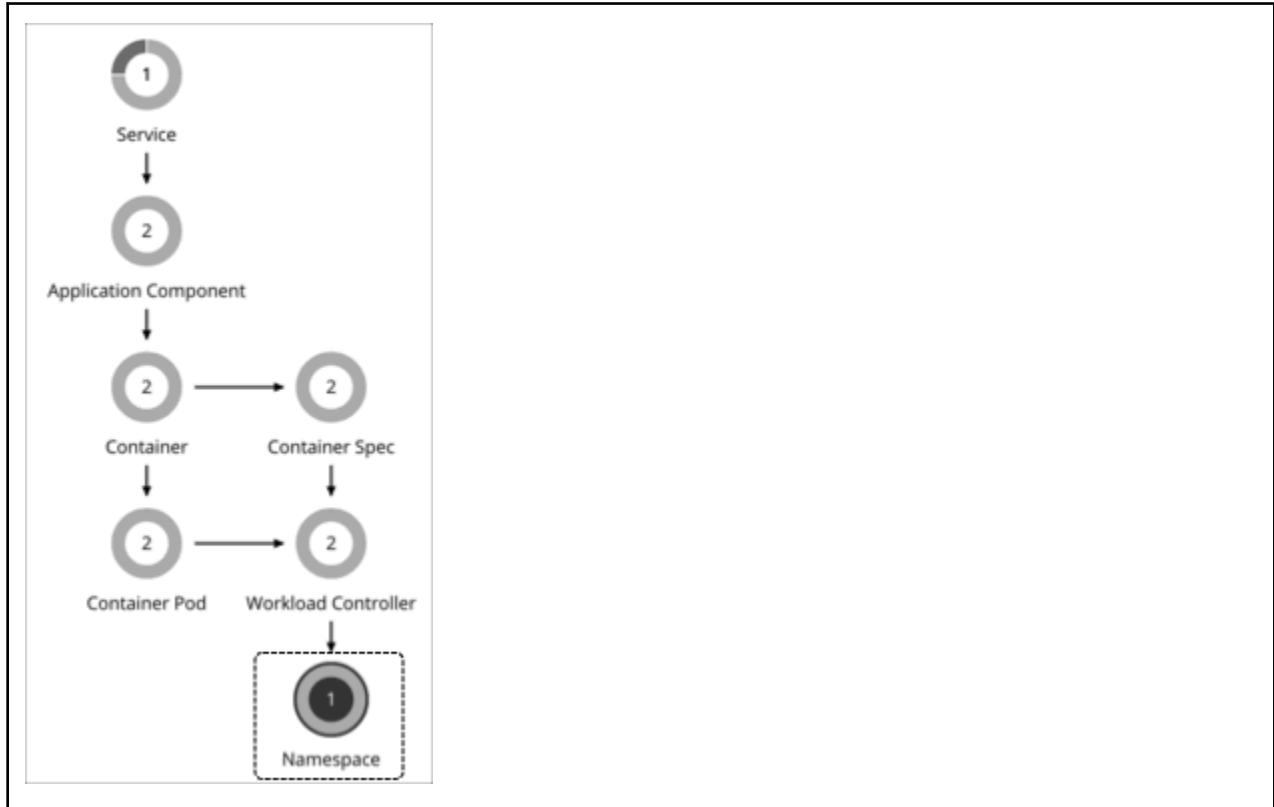
For more information, see [Creating Placement Policies](#).



## Namespace

A namespace is a logical pool of resources in a Kubernetes environment that manages workloads based on specific requirements or business needs. For example, administrators can pool resources for different organizations within the enterprise, and assign different policies to each pool.

### Synopsis



### Synopsis

|                     |                         |
|---------------------|-------------------------|
| Budget:             | N/A                     |
| Provides:           | N/A                     |
| Consumes:           | N/A                     |
| Discovered through: | Kubeturbo Mediation Pod |

### Labels and Annotations

Turbonomic discovers namespace labels and annotations as tag properties. You can filter namespaces by labels or annotations when you use Search or create Groups.

### Monitored Resources

A namespace can include the following compute resource quotas:

- VMem Request Quota

VMem Request Quota is the total amount of virtual memory request for all pods allocated to the namespace against the namespace quota.

- VCPU Request Quota

VCPU Request Quota is the total amount of virtual CPU request (in mCores) for all pods allocated to the namespace against the namespace quota.

- VMem Limit Quota

VMem Limit Quota is the total amount of virtual memory limit for all pods allocated to the namespace against the namespace quota.

- VCPU Limit Quota

VCPU Limit Quota is the total amount of virtual CPU limit (in mCores) for all pods allocated to the namespace against the namespace quota.

When configured, these quotas define the capacity for the given namespace. Turbonomic monitors actual utilization of these quotas against cluster capacity.

Points to consider:

- When you scope to a namespace in the supply chain, you can see utilization data in the **Capacity and Usage** and **Namespace Multiple Resources** charts. With this data, you can understand how pods running in the namespace are consuming resources.
- The Capacity and Usage chart shows *Capacity* as the namespace quotas. *Used* values are the sum of resource limits and/or requests set for all pods in the namespace.

| COMMODITY              | CAPACITY   | USED       | UTILIZATION |
|------------------------|------------|------------|-------------|
| Memory Request Quota ⓘ | 640 MB     | 640 MB     | 100%        |
| CPU Limit Quota ⓘ      | 500 mCores | 500 mCores | 100%        |
| Memory Limit Quota ⓘ   | 1.25 GB    | 1.25 GB    | 100%        |
| CPU Request Quota ⓘ    | 250 mCores | 100 mCores | 40%         |
| Virtual Memory Request | 90.99 GB   | 640 MB     | 0.69%       |

For a namespace that does not have defined quotas, *Capacity* for the commodity is infinite (as shown in the image below). *Used* values are the sum of resource limits and/or requests set for all pods in the namespace. If these are not set, *Used* value is 0 (zero).

| Capacity and Usage     |           |            |             |
|------------------------|-----------|------------|-------------|
| openshift-sdn          |           |            |             |
| COMMODITY              | CAPACITY  | USED       | UTILIZATION |
| Memory Request Quota ⓘ | ∞         | 4.99 GB    | 0%          |
| CPU Request Quota ⓘ    | ∞         | 1.03 Cores | 0%          |
| Memory Limit Quota ⓘ   | ∞         | 0 KB       | 0%          |
| CPU Limit Quota ⓘ      | ∞         | 0 mCores   | 0%          |
| Virtual Memory Request | 192.04 GB | 4.99 GB    | 2.6%        |

SHOW ALL >

**Note:**

If you download the data in the chart, the downloaded file shows infinite capacities as unusually large values (for example, 1,000,000,000 cores instead of the ∞ symbol).

- To see which namespaces use the most cluster resources, set the scope to a container cluster and see the **Top Namespaces** chart. You can use the data in the chart for showback analysis.
- When you run Optimize Container Cluster plans, Turbonomic can calculate increased namespace quotas in the plan results. For more information, see [Optimize Container Cluster Plan](#).

## Namespace Actions

### Resize Quota

Turbonomic treats quotas defined in a namespace as constraints when making container resize decisions. If existing container actions would exceed the namespace quotas, Turbonomic recommends actions to resize up the affected namespace quota.

Note that Turbonomic does not recommend actions to resize *down* a namespace quota. Such an action reduces the capacity that is already allocated to an application. The decision to resize down a namespace quota should include the application owner.

When you have a recommendation to resize namespace quotas, Turbonomic blocks execution of the resize actions for the affected Workload Containers. The action details show these blocked actions in the **Related Actions** list.

### Action Details

Resize up VMem Limit Quota for Namespace [quota-test-with-down](#) from 3.4 GB to 4 GB  
VMem Limit Congestion in Related Workload Controller

TAGS

kubernetes.io/metadata.name: quota-test-with-down

**NAMESPACE - IMPACT FROM ALL ACTIONS**

[quota-test-with-down](#)

|                                                                        |                                                                             |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <p>MEMORY LIMIT QUOTA</p> <p>100 % → 84.8 % ↓</p> <p>3.4 GB → 4 GB</p> | <p>CPU LIMIT QUOTA</p> <p>100 % → 88.9 % ↓</p> <p>3.2 Cores → 3.6 Cores</p> |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------|

STATE

Action acceptance is blocked by policy or system.  
Acceptance mode is Recommend.

**RELATED ACTIONS**

BLOCKING

Resize VCPU Limit,VMem Limit for Workload Controller [cpu-quota-3](#) in EA - Advanced Engineering  
Resize VCPU Limit,VMem Limit for Workload Controller [cpu-quota-1](#) in EA - Advanced Engineering

**Note:**

For more information about execution of container resize actions via Workload Controllers, see [Workload Controller Actions](#).

**Temporary Increases in Namespace Quotas**

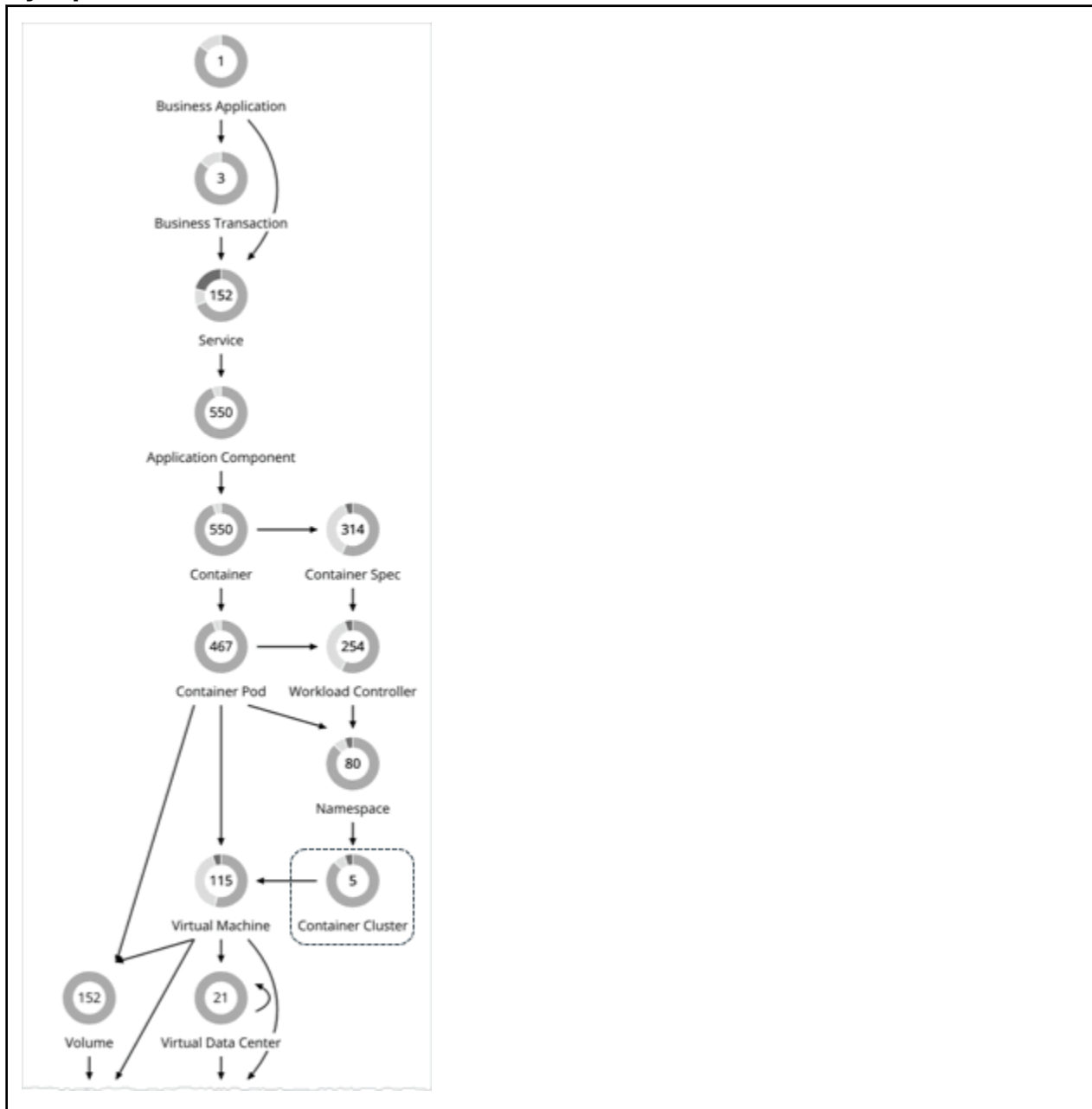
When executing a resize action on a Workload Controller in a namespace with a defined quota, Turbonomic can increase the namespace quota temporarily to accommodate new replicas.

When the execution of a resize action requires new replicas, a Kubernetes workload usually has a rolling update strategy defined. Even if the namespace quota is sufficient for the resource requirement of the new replicas, there is a chance that the quota is not enough to accommodate both the old and new replicas, as required by the rolling update. In this case, Kubeturbo calculates and then increases the quota based on the resources required by both the old and new replicas. Kubeturbo reverts the quota to its original size after the new replicas have been scheduled on a node.

**Container Cluster**

A Container Cluster is a Kubernetes cluster that Turbonomic discovers through Kubeturbo. With this entity type, Turbonomic can fully link the entire container infrastructure with the underlying nodes, and then present all actions on containers and nodes in a single view. This gives you full visibility into the actions that impact the health of your container environment.

## Synopsis



### Synopsis

|                     |                         |
|---------------------|-------------------------|
| Budget:             | N/A                     |
| Provides:           | N/A                     |
| Consumes:           | N/A                     |
| Discovered through: | Kubeturbo Mediation Pod |

### Integration with Git-based Software via Argo CD

Argo CD represents each application as a Custom Resource (CR) named ArgoCD Application. If Argo CD is installed in a given cluster, Turbonomic can perform the following:

- Identify ArgoCD Applications in the cluster and represent them as Business Applications in the Turbonomic supply chain.
- Identify the container cluster resources consumed by the applications and stitch them to the relevant Business Application in the supply chain.
- Discover details for the Git repository that acts as the 'source of truth' for Argo CD, including the Git repository, branch, and file path for the ArgoCD Applications. Turbonomic uses these details to drive container resize actions and sync the files updated by resizes with the Git repository. Currently, this integration works for applications based on plane YAML files, with GitHub or GitLab as the source of truth.

Be sure to [install Argo CD](#) in the cluster managed by Turbonomic. For instructions on integrating Turbonomic with GitHub or GitLab via Argo CD, visit this [page](#).

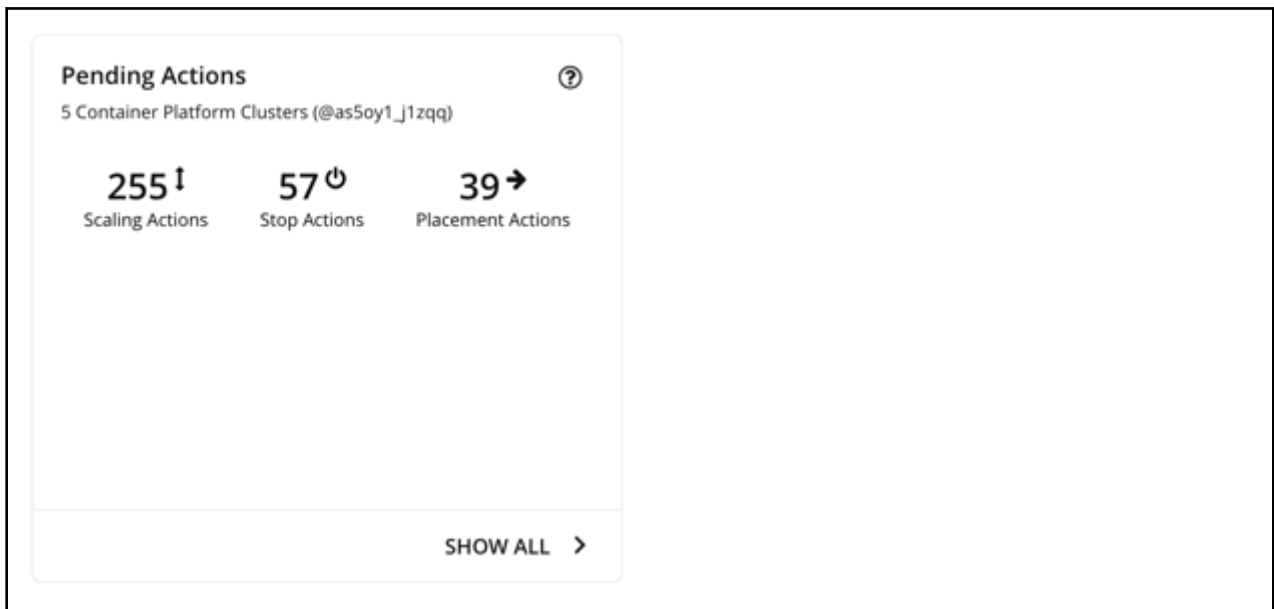
## Monitored Resources

Turbonomic monitors resources for the containers, pods, nodes (VMs), and volumes in a cluster.

## Actions

### None

Turbonomic does not recommend actions for a Container Cluster. Instead, it recommends actions for the containers, pods, nodes (VMs), and volumes in the cluster. Turbonomic shows all of these actions when you scope to a Container Cluster and view the Pending Actions chart.



For actions on nodes:

- For actions to suspend or provision nodes in the public cloud, Turbonomic includes cost information (investments or savings) attached to those actions. Note that Turbonomic generates these actions *not* to optimize costs, but to assure performance and efficiency for your container infrastructure. Turbonomic reports costs to help you track your cloud spend.

To view cost information, set the scope to a cluster in the public cloud and view the Necessary Investments or Potential Savings charts. You can also set the scope to the global cloud environment to see total costs, or to individual container clusters or nodes.

- For VMs/nodes that make up an Azure Kubernetes Service (AKS) cluster, you can manually execute recommended VM Provision and VM Suspend actions. This adjusts the count of nodes in a given node pool, where Provision raises the node count, and Suspend lowers it. You can execute these actions if the cluster is also discovered through an Azure target (along with the KubeTurbo target).

- Node pools and machine sets are ways to deploy and scale compute resources for Kubernetes services hosted in the public cloud and the Red Hat OpenShift 4.x container platform on any infrastructure.

For Kubernetes services in the public cloud, Turbonomic uses default labels with the following patterns to discover the node pool types within each cluster:

- Azure Kubernetes Service (AKS): `agentpool`
- Amazon Elastic Kubernetes Service (EKS):  
`//alpha.eksctl.io/nodegroup-name`  
`eks.amazonaws.com/nodegroup`
- Google Kubernetes Engine (GKE): `cloud.google.com/gke-nodepool`

For Red Hat OpenShift 4.x, Turbonomic creates node pools based on machine sets.

For both discovered and auto-created node pools, Turbonomic aggregates and visualizes actions for all the nodes in a pool to help you identify performance issues and optimization opportunities at the node pool level. Use the Top Node Pools chart to see actions and detailed information. By default, this chart displays when you set the scope to your global environment and then click the Container Cluster entity in the supply chain.

| Name                                            | Node Count | Potential Savings | Potential Investments | Actions    |
|-------------------------------------------------|------------|-------------------|-----------------------|------------|
| NodePool-machineset-ocp47demo-2v5jc-worker-us   | 6          | \$165.11/mo       | \$165.11/mo           | 26 ACTIONS |
| NodePool-eks-cluster-ng2-Kubernetes-EKS-withWin | 2          | \$8.69/mo         | N/A                   | 1 ACTION   |
| NodePool-agentpool-Kubernetes-AKS               | 4          | \$0.00/mo         | N/A                   | no actions |
| NodePool-ami-005fb2dc84caa293d-Kubernetes-EK!   | 2          | \$0.00/mo         | N/A                   | no actions |
| NodePool-ami-0a99721a12001ebd4-Kubernetes-EK    | 2          | \$0.00/mo         | N/A                   | no actions |
| NodePool-bar-Kubernetes-DC11-PT-KBs             | 1          | \$0.00/mo         | N/A                   | no actions |
| NodePool-eks-cluster-ng1-Kubernetes-EKS-withWin | 1          | \$0.00/mo         | N/A                   | no actions |

The chart shows the number of nodes and aggregated actions for each node pool. For node pools in the public cloud, the chart also shows the costs you would incur if you provision nodes and then scale their volumes, or the savings you would realize if you suspend nodes. To view individual actions, click the button under the Actions column. To see more details, including the full list of nodes for each pool, click the node pool name.

You can automate the execution of these actions through Turbonomic with Red Hat OpenShift 4.x Machine Operator, or via an Action Script. You can also manually execute node actions for AKS, EKS, or GKE via the cloud provider.

**Note:**

The following capabilities will be introduced in a future release:

- Actions to provision or suspend nodes via a plan simulation

- Policies for node pools
- Execution of node actions for AKS, EKS and GKE through Turbonomic

## Cluster Health

To assess the health of each cluster, see the **Top Container Platform Clusters** chart in the predefined Container Platform Dashboard.

For each cluster, the chart shows the sum of resources used by containers and the underlying nodes. Click the **Actions** button to see a list of pending actions.

| Container Cluster      | Health | Virtual CPU Used        | Virtual CPU Request      | Virtual Memory Used   | Virtual Memory Request | Actions     |
|------------------------|--------|-------------------------|--------------------------|-----------------------|------------------------|-------------|
| Kubernetes-PT-AKS      |        | 8.2 Cores (41%) ↓ 0.31% | 13.01 Cores (68%)        | 20.58 GB (28%) ↓ 0%   | 21.66 GB (40%)         | 60 ACTIONS  |
| Kubernetes-DC11-PT-KBs |        | 6.78 Cores (28%) ↓ 3%   | 11.46 Cores (50%)        | 24.67 GB (26%) ↑ 0.1% | 12.27 GB (13%)         | 139 ACTIONS |
| Kubernetes-OCP47-AWS   |        | 13.36 Cores (26%) ↑ 2%  | 12.87 Cores (27%) ↑ 3%   | 90.84 GB (45%) ↓ 13%  | 72.4 GB (38%) ↑ 0.69%  | 217 ACTIONS |
| Kubernetes-OKD-311     |        | 4.07 Cores (16%) ↓ 9%   | 8.86 Cores (35%) ↑ 1%    | 27.94 GB (30%) ↑ 23%  | 21.35 GB (23%) ↑ 0.92% | 119 ACTIONS |
| Kubernetes-ocp-wdc02   |        | 10.02 Cores (21%) ↓ 17% | 10.6 Cores (22%) ↓ 0.92% | 208.77 GB (55%) ↓ 24% | 103.17 GB (30%) ↑ 2%   | 319 ACTIONS |
| Kubernetes-OCP43-AWS   |        | 3.32 Cores (21%) ↑ 17%  | 7.41 Cores (51%) ↓ 0%    | 23.72 GB (38%) ↑ 2%   | 15.53 GB (26%) ↓ 0%    | 16 ACTIONS  |
| Kubernetes-Turbonomic  |        | 5.11 Cores (64%) ↑ 27%  | 0.95 Cores (12%)         | 61.83 GB (49%) ↑ 5%   | 18.03 GB (14%)         | 73 ACTIONS  |

The **Top Namespaces** chart shows the namespaces that use the most cluster resources. You can use the data in the chart for showback analysis.

| Namespace            | H. Container Cluster | Virtual CPU Used        | CPU Request Quota | CPU Limit Quota  | Virtual Memory Used    | Memory Request Quota | Memory Limit Quota | Actions    |
|----------------------|----------------------|-------------------------|-------------------|------------------|------------------------|----------------------|--------------------|------------|
| demoapp              | OKD-311              | 152 mCores (1%) ↓ 0.19% | 100 mCores (0%)   | 200 mCores (0%)  | 1.91 MB (0%)           | 10 MB (0%)           | 20 MB (0%)         | 72 ACTIONS |
| action-merge-test    | OKD-311              | 16 mCores (0%) ↓ 5%     | 100 mCores (0%)   | 200 mCores (0%)  | 153.23 MB (0%) ↑ 0.08% | 260 MB (0%)          | 400 MB (0%)        | 75 ACTIONS |
| action-merge-test2   | OKD-311              | 202 mCores (1%) ↓ 0.07% | 90 mCores (0%)    | 202 mCores (0%)  | 2.2 MB (0%) ↑ 4%       | 260 MB (0%)          | 400 MB (0%)        | 78 ACTIONS |
| turbo-operator-arsen | OCP47-AWS            | 92 mCores (0%) ↑ 279%   | 200 mCores (0%)   | 1 Cores (0%)     | 202.93 MB (0%) ↑ 6%    | 512 MB (0%)          | 1 GB (0%)          | 5 ACTIONS  |
| provoit              | DC11-PT-KBs          | 442 mCores (2%) ↓ 0.09% | 606 mCores (15%)  | 5.78 Cores (72%) | 585.42 MB (1%)         | 2.44 GB (24%)        | 20.88 GB (42%)     | 84 ACTIONS |
| instana-agent        | OCP47-AWS            | 1.27 Cores (2%) ↑ 1%    | 2.5 Cores (0%)    | 7.5 Cores (0%)   | 2.58 GB (1%) ↓ 4%      | 2.5 GB (0%)          | 2.5 GB (0%)        | 17 ACTIONS |
| aqiql-cpu-throttling | DC11-PT-KBs          | 757 mCores (3%) ↑ 0.18% | 600 mCores (0%)   | 1.35 Cores (0%)  | 23.71 MB (0%)          | 50 MB (0%)           | 100 MB (0%)        | 78 ACTIONS |

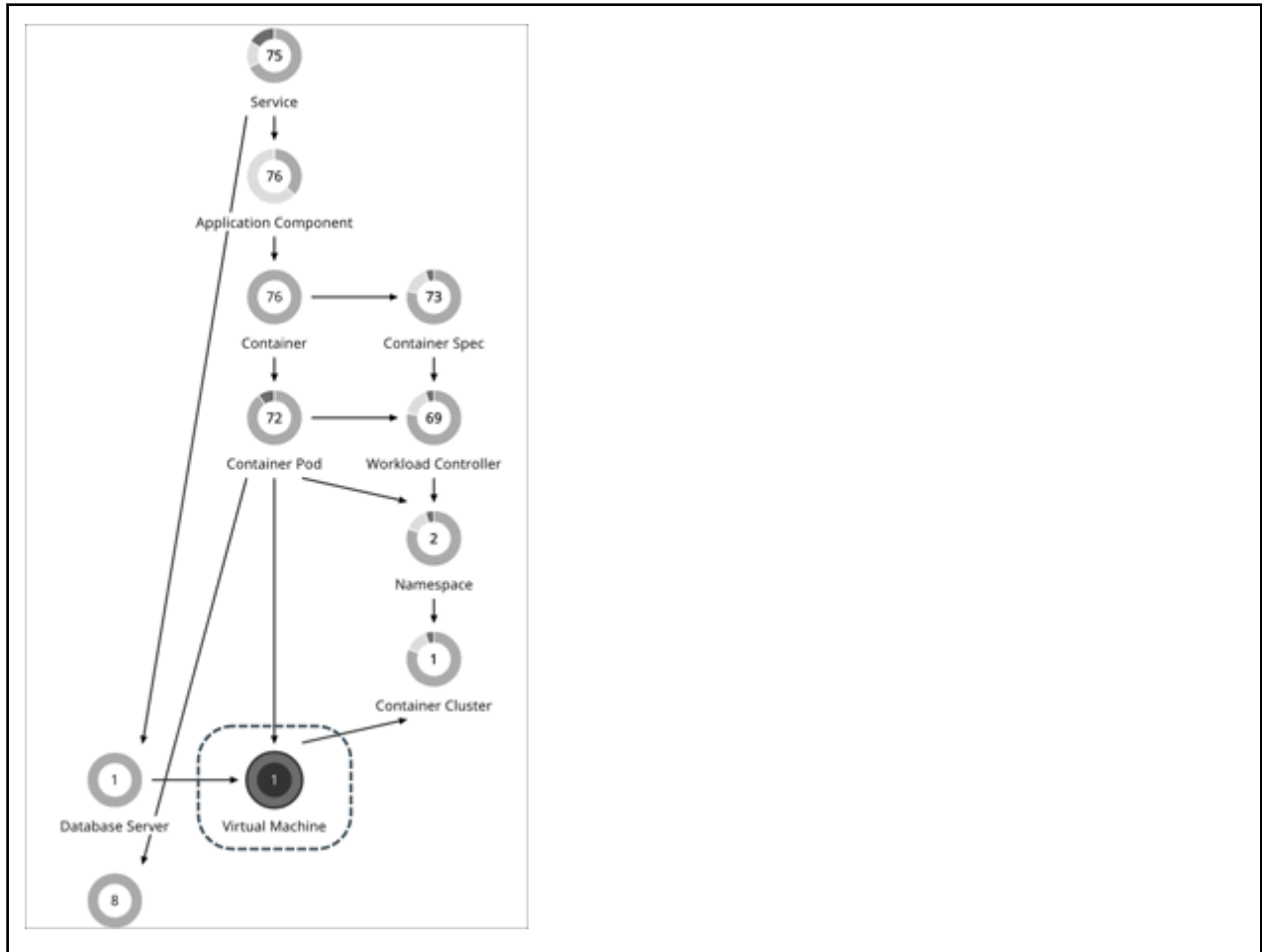
## Virtual Machine (Kubernetes Node)

In Kubernetes environments, a node is a virtual or physical machine that contains the services necessary to run pods. Turbonomic represents nodes as Virtual Machine entities in the supply chain.



Turbonomic can discover node roles and Master Nodes. It creates policies to keep nodes of the same role on unique host or Availability Zone providers, and policies to disable suspension of Master Nodes. Turbonomic also discovers and displays Node Pools, and Red Hat OpenShift Machine Sets.

## Synopsis



## Synopsis

|                     |                                   |
|---------------------|-----------------------------------|
| Provides:           | Resources to pods                 |
| Consumes:           | Resources from container clusters |
| Discovered through: | Kubeturbo Mediation Pod           |

## Monitored Resources

Turbonomic monitors the following resources for nodes that host Kubernetes pods. These resources are monitored along with the resources from the infrastructure probes, such as vCenter or a public cloud mediation probe.

- VMem  
VMem is the virtual memory currently used by all containers on the node. The capacity for this resource is the Node Physical capacity.
- VCPU

VCPU is the virtual CPU currently used by all containers on the node. The capacity for this resource is the Node Physical capacity.

- **Memory Request Allocation**

Memory Request Allocation is the memory available to the node to support the ResourceQuota request parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- **CPU Request Allocation**

CPU Request Allocation is the CPU available to the node to support the ResourceQuota request parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- **Virtual Memory Request**

Virtual Memory Request is the memory currently guaranteed by all containers on the node with a Memory Request. The capacity for this resource is the Node Allocatable capacity, which is the amount of resources available for pods and can be less than the physical capacity.

- **Virtual CPU Request**

Virtual CPU Request is the CPU currently guaranteed by all containers on the node with a CPU Request. The capacity for this resource is the Node Allocatable capacity, which is the amount of resources available for pods and can be less than the physical capacity.

- **Memory Allocation**

Memory Allocation is the memory ResourceQuota limit parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

- **CPU Allocation**

CPU Allocation is the CPU ResourceQuota limit parameter for a given namespace (Kubernetes namespace or Red Hat OpenShift project).

## Actions

Turbonomic can recommend the following actions:

- **Provision**

Provision nodes to address workload congestion or meet application demand.

- **Suspend**

Suspend nodes after you have consolidated pods or defragmented node resources to improve infrastructure efficiency.

- **Reconfigure**

Reconfigure nodes that are currently in the NotReady state.

### Note:

For nodes in the public cloud, Turbonomic reports the cost savings or investments attached to node and provision actions. For example, you can see the additional costs you would incur if you provision nodes and then scale their volumes, or the savings you would realize if you suspend nodes. Note that performance and efficiency are the drivers of these actions, *not* cost. Cost information is included to help you track your cloud spend. For this reason, you will *not* see cost-optimization actions, including recommendations to re-allocate discounts or delete unattached volumes.

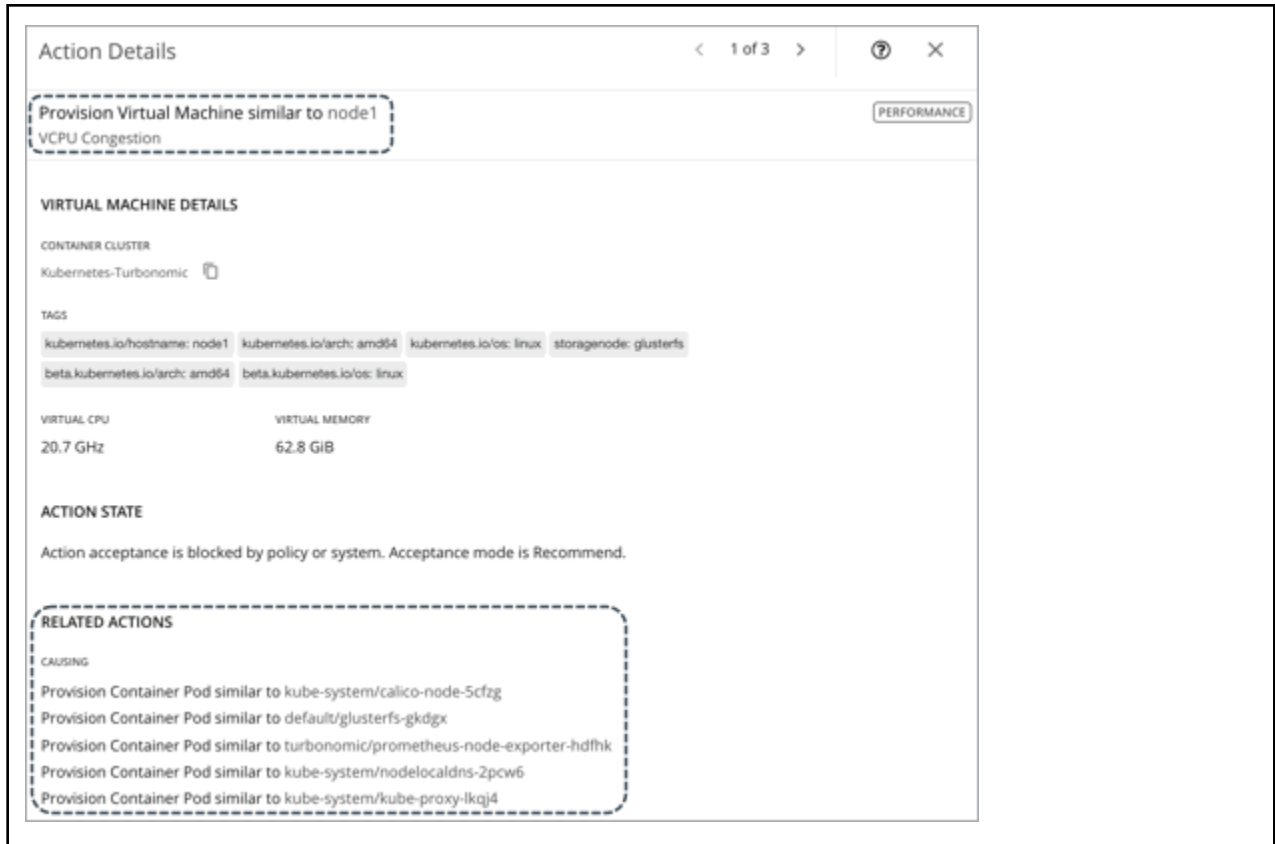
To view cost information, set the scope to a node and see the Necessary Investments and Potential Savings charts. You can also set the scope to a [container cluster](#) or the global cloud environment to view aggregated cost information.

## Node Provision Actions

When recommending node provision actions, Turbonomic also recommends pod provision actions that reflect the projected demand from required DaemonSet pods, and respects the maximum number of pods

allowed for a node. This ensures that any application workload can be placed on the new node and stay within the desired range of vMem/vCPU usage, vMem/vCPU request, and number of consumers.

The action details for a node provision action show the related DaemonSet pods that are required for the node to run. Click a pod name to set it at your scope.



Turbonomic treats [static pods](#) as DaemonSets for the purpose of provisioning nodes. Because a static pod provides a node with a specific capability, it is controlled by the node and is not accessible through the API server. If a node to be provisioned requires a static pod, Turbonomic generates actions to provision the node and the corresponding static pod.

## Node Suspension Actions

When recommending node suspension actions, Turbonomic also recommends suspending the DaemonSet pods that are no longer required to run the suspended nodes.

The action details for a node suspension action show the related DaemonSet pods that are no longer needed to run the suspended nodes. Click a pod name to set it at your scope.

**Action Details** < 1 of 2 > ? X

**Suspend Virtual Machine** ocp47demo-2v5jc-worker-us-east-2a-ctc7p  
Improve infrastructure efficiency ↓ \$280.00/mo SAVINGS

**VIRTUAL MACHINE DETAILS**

| NAME                      | ID                  | AGE        | ACCOUNT              |
|---------------------------|---------------------|------------|----------------------|
| ocp47demo-2v5jc-worker... | I-0ae58b02dc839bcd6 | 30+ day(s) | Advanced Engineering |

**CONTAINER CLUSTER**  
Kubernetes-AWS-0CP-47

**TAGS**  
kubernetes.io/cluster/ocp47demo-2v5jc: owned

| VIRTUAL CPU | VIRTUAL MEMORY | NUMBER OF CONSUMERS |
|-------------|----------------|---------------------|
| 25.2 GHz    | 30.6 GiB       | 250                 |

**COST IMPACT**

| ON-DEMAND RATE | ON-DEMAND COST |
|----------------|----------------|
| \$0.384/h      | \$0.384/h      |

**STATE**  
Action acceptance is blocked by policy or system. Acceptance mode is Recommend.

**RELATED ACTIONS**

CAUSING

- Suspend Container Pod openshift-sdn/sdn-rc94c
- Suspend Container Pod openshift-image-registry/node-ca-whzhh
- Suspend Container Pod openshift-dns/dns-default-qxr8h

Turbonomic treats static pods as DaemonSets for the purpose of suspending nodes. Because a static pod provides a node with a specific capability, it is controlled by the node and is not accessible through the API server. If the only workload type left on a node is a static pod, Turbonomic generates actions to suspend the node and the corresponding static pod.

## Node Reconfigure Actions

Turbonomic generates node reconfigure actions to notify you of nodes that are currently in the NotReady state.

A reconfigure action should change a node's state to Ready so that Turbonomic can begin to monitor the health of the node and the associated container pods. This action is read-only and must be executed outside Turbonomic. As part of action execution, you might need to restart the node or the kubelet agent on the node.

### Note:

Turbonomic treats a node as a VM under certain circumstances. For example, it treats a node in vCenter as a VM that can move to a different host if the current host is congested. This means that for a NotReady node in vCenter, it is possible to see a VM move action along with the expected node reconfigure action. Both actions are valid and safe to execute since they achieve two different and non-conflicting results.

For each Kubernetes cluster, Turbonomic creates an auto-generated group of NotReady nodes. To view all the auto-generated groups, go to Search, select Groups, and then type `notready` as your search keyword. Click a group to view the individual nodes and the pending reconfigure actions.

Search  
Search within your infrastructure

Accounts

App Component Specs

Application Components

Billing Families

Business Applications

Business Transactions

Business Users

Chassis

Clusters

Container Platform Clusters

Container Pods

Container Specs

Containers

Data Centers

Database Servers

Databases

Desktop Pools

Disk Arrays

Folders

Groups

Hosts

notready

ADD FILTER

|                                                  |                                 | NAME   | ↑ |
|--------------------------------------------------|---------------------------------|--------|---|
| NotReady Nodes [Kubernetes-ae-cluster-1]         | 2                               | Static | > |
| <small>On-Prem   Kubernetes-ae-cluster-1</small> | <small>Virtual Machines</small> |        |   |
| NotReady Nodes [Kubernetes-Hybrid]               | 2                               | Static | > |
| <small>Hybrid   Kubernetes-Hybrid</small>        | <small>Virtual Machines</small> |        |   |

When you examine a pending reconfigure action, you can click the link in the 'Entities Impacted by this Node' section to view a list of impacted pods.

Action Details

Reconfigure Virtual Machine ae-cluster1-node-group-afdd79db90

The node is in a NotReady status

**VIRTUAL MACHINE DETAILS**

NAME  
ae-cluster1-node-group-afdd79db90

| VMEM PERCENTILE | VCPU PERCENTILE |
|-----------------|-----------------|
| 31 %            | 11 %            |
| 8 GB            | 4.8 GHz         |

STATE  
Action acceptance is blocked by policy or system. Acceptance mode is Recommend.

**ENTITIES IMPACTED BY THIS NODE**

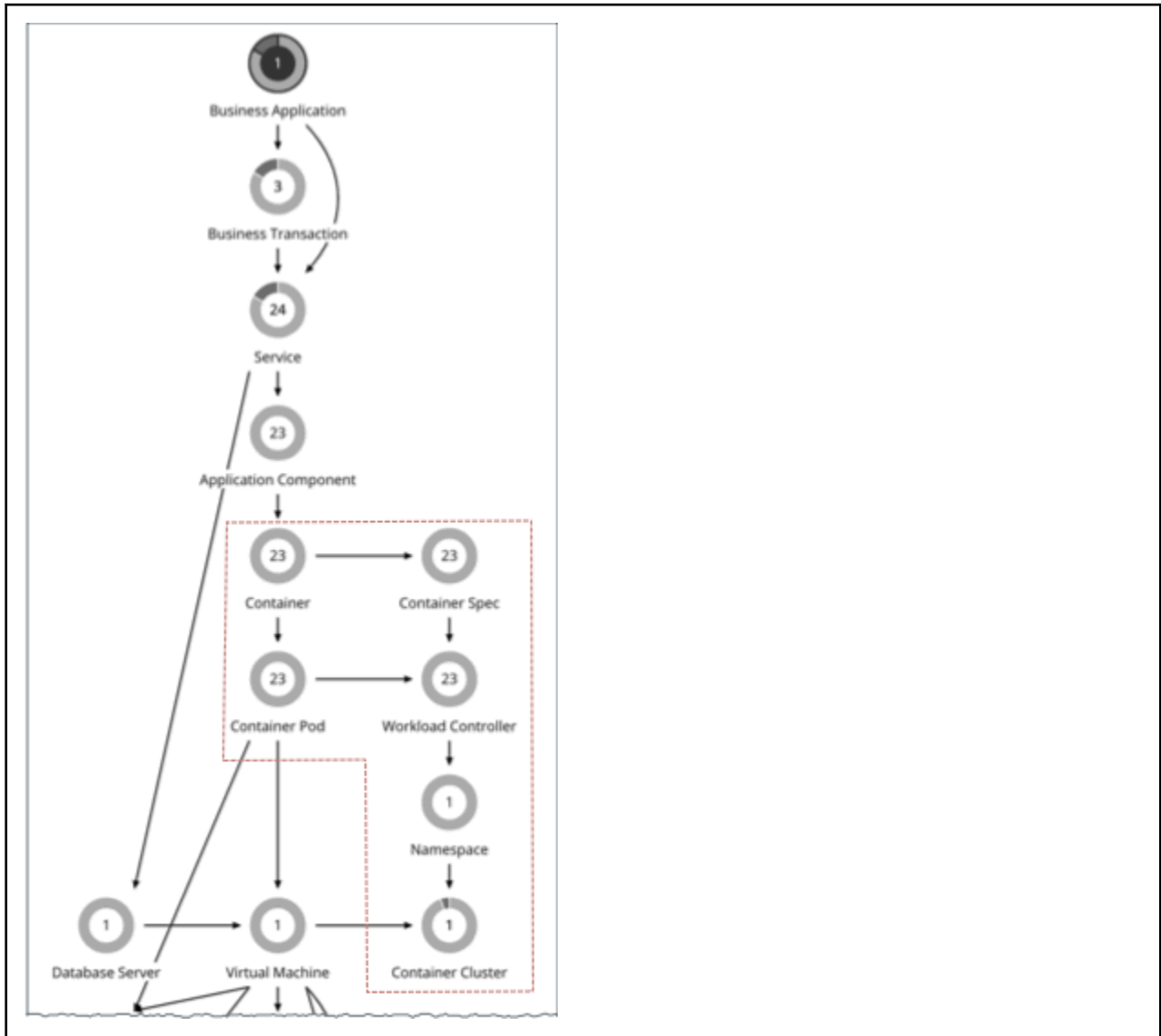
Reconfiguring this VM might activate the container pods that are currently in an unknown state.

[View List of Container Pods in Unknown State](#)

These pods are in the Unknown state and are not controllable. In the supply chain and in the list of container pods, these pods display with a gray color to help you differentiate them from other pods.

## Kubernetes CPU Metrics

To meet user requirements and align with Kubernetes specifications, Turbonomic uses millicore (mCore) as the base unit for CPU metrics for your Kubernetes platform.



These include metrics for the following CPU-related commodities:

- vCPU
- vCPU Request
- vCPU Limit Quota
- vCPU Request Quota

Turbonomic displays these commodities in charts, actions, policies, and plans. For example:

- In the Capacity and Usage chart for container platform entities, *capacity* and *used* values for CPU-related commodities are shown in mCores.
- In the supply chain, when you scope to a Workload Controller to view pending [resize actions](#) for a container, you will see utilization and resize values in mCores.
- When you create [Container Spec policies](#), resize thresholds and increment constants for CPU-related commodities are set in mCores.

- For an Optimize Container Cluster plan, the plan results for CPU-related commodities are shown in mCores.

For nodes (VMs) and Application Components:

- For nodes stitched to your Kubernetes platform, the base unit for *vCPU Request* is also mCore, since this commodity is provided only to Kubernetes.
- For both nodes and Application Components (standalone or stitched to your Kubernetes platform), the base unit for *vCPU* is MHz, since this is a generic commodity. For example, when you view a pod move action, vCPU metrics for the current and destination nodes for the pod are expressed in MHz.

The following table summarizes the base units of CPU measurement that Turbonomic uses.

| Entity                | CPU Commodity |              |                  |                    |
|-----------------------|---------------|--------------|------------------|--------------------|
|                       | vCPU          | vCPU Request | vCPU Limit Quota | vCPU Request Quota |
| Container             | mCore         | mCore        | mCore            | mCore              |
| Container Spec        | mCore         | mCore        | N/A              | N/A                |
| Workload Controller   | N/A           | N/A          | mCore            | mCore              |
| Container Pod         | mCore         | mCore        | mCore            | mCore              |
| Namespace             | mCore         | mCore        | mCore            | mCore              |
| Container Cluster     | mCore         | mCore        | N/A              | N/A                |
| Node (VM)             | MHz           | mCore        | N/A              | N/A                |
| Application Component | MHz           | N/A          | N/A              | N/A                |

This feature is available starting in version 8.2.5. For customers updating to version 8.2.5 or later:

- This feature does *not* require you to update your Kubeturbo image after the update.
- For time series charts, metrics generated *after* the update are actual mCore values, but pre-update metrics are the same (unconverted) values in MHz displayed in mCore units. This results in unexpected data in charts immediately after the update.

For example:

If vCPU Limit for a Container Spec was resized from 1300 MHz to 1200 MHz *before* you updated Turbonomic, data points in charts correctly show these values in MHz.

Immediately after the update:

- When you view the Virtual CPU chart for the Container Spec, Turbonomic will show a capacity value of 1200 mCores (which is 1200 MHz in reality) for the last data point *before* the update, and the equivalent value of 500 mCores for the first data point *after* the update. This gives the impression of a resize down action between the data points, even if no such action was executed.
- Assume Turbonomic recommends an action to resize VCPU Limit for the Container Spec from 500 to 700 mCores. When you view the details for this action via the associated Workload Controller, the time series chart will show unexpected data.
- For the actual recommended action, the data point shows current capacity as 1200 mCores, instead of 500 mCores. The new value after executing the action correctly shows as 700 mCores.

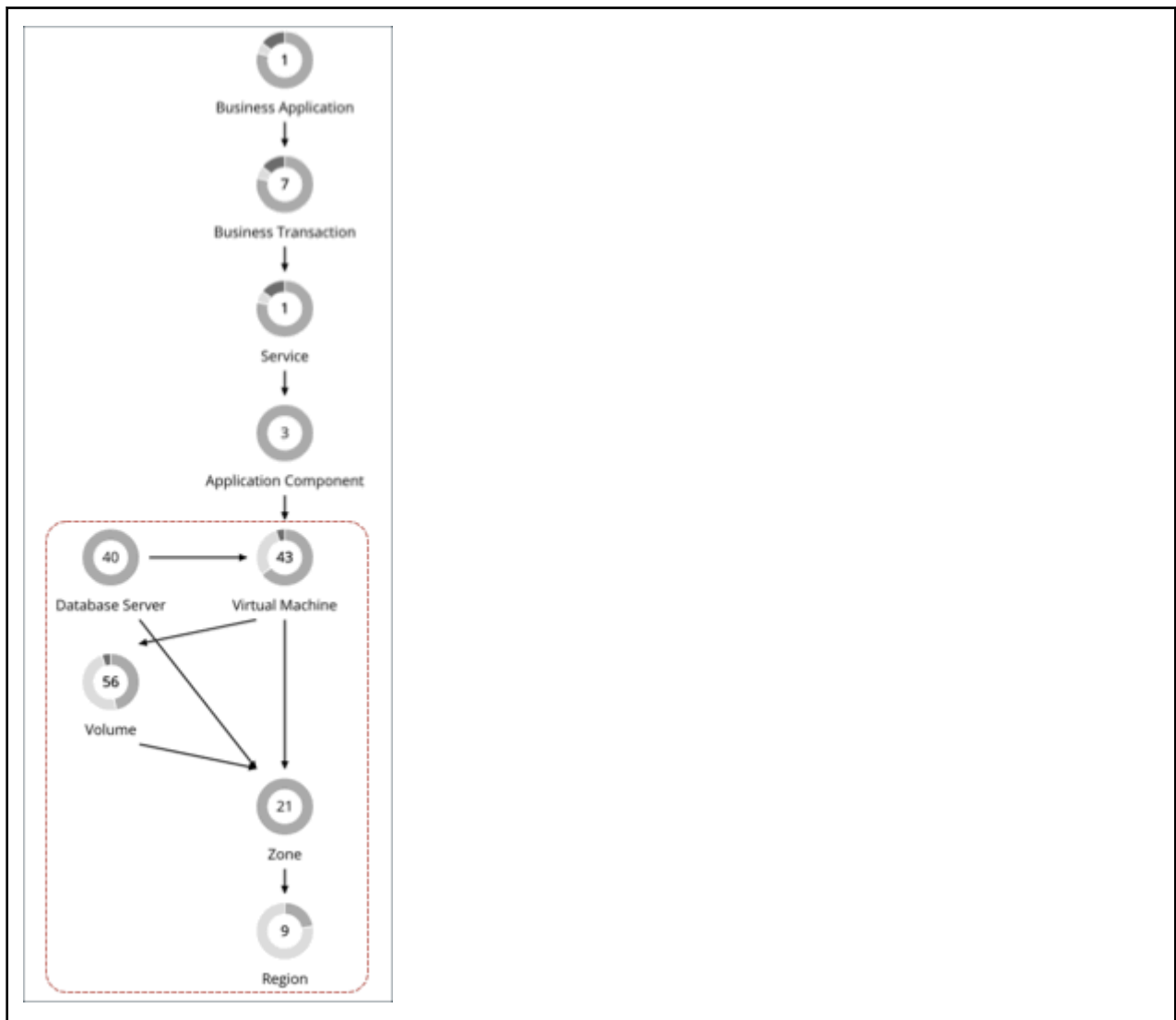
- For the last resize action before the update, the data point shows the same MHz values (1300 and 1200), but in mCore units.
- One day after the update, a new data point displays in the chart, indicating that capacity was resized from 1200 mCores to 500 mCores, even if no actual resize action was executed.

Over time, data points with unexpected values will begin to fall out of range and newer data points will reflect actual mCore values.

- For increment constants in Container Spec policies, the default value of 100 remains unchanged, but the unit changes from MHz to mCores. This means that each resize action will now increase or decrease capacity by 100 mCores, instead of 100 MHz.

## Entity Types - Cloud Infrastructure

Turbonomic discovers and monitors the entities that make up your cloud infrastructure, and recommends actions to assure application performance at the lowest possible cost.



### Virtual Machine (Cloud)

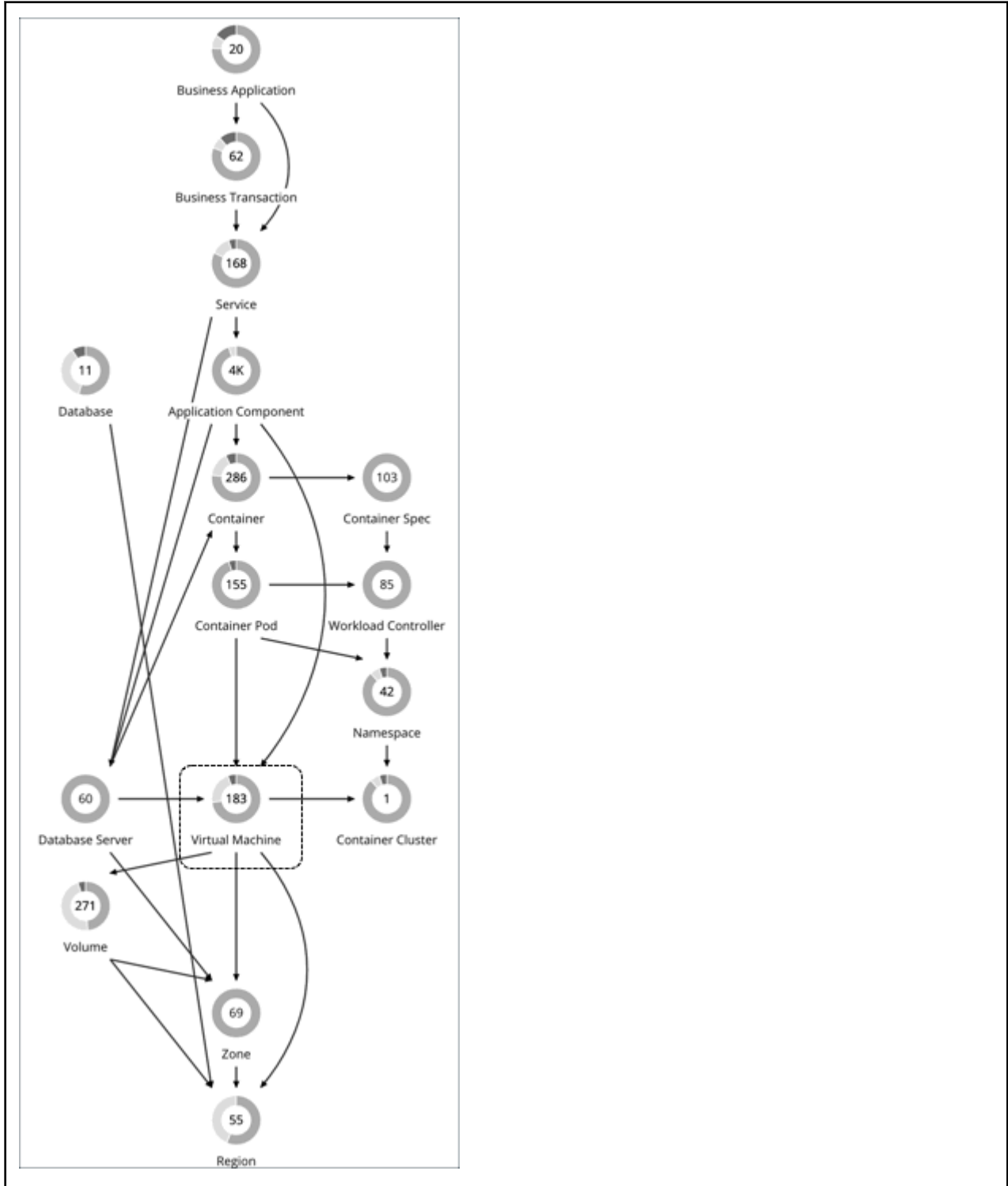
A virtual machine (VM) is a software emulation of a physical machine, including OS, virtual memory and CPUs, and network ports. VMs host applications, or they provide resources to container platforms.

**Note:**



Kubernetes nodes are represented as Virtual Machines in the Turbonomic supply chain. For details about nodes, see [Virtual Machine \(Kubernetes Node\)](#).

## Synopsis



## Synopsis

Provides:

Resources for applications to use

## Synopsis

|                     |                                       |
|---------------------|---------------------------------------|
| Consumes:           | Resources from cloud zones or regions |
| Discovered through: | Cloud targets                         |

## Monitored Resources

For details about monitored resources for cloud VMs, see the following topics:

- [AWS Monitored Resources](#)
- [Azure Monitored Resources](#)
- [GCP Monitored Resources](#)

## Cloud VM Actions

For details about cloud VM actions, see the following topics:

- [Actions for AWS VMs](#)
- [Actions for Azure VMs](#)
- [Actions for GCP VMs](#)

## Actions for AWS VMs

Turbonomic can recommend the following actions:

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

See additional information for scale actions below.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing RI coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

If you need more capacity, then Turbonomic will recommend actions to purchase additional RIs.

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

- **Stop and Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

For details, see [Parking: Stop or Start Cloud Resources](#).

## Controlling Scale Actions for AWS VMs

For scale actions, you can create policies to control the scale actions that Turbonomic recommends. In those policies, choose from the following options:

- Cloud Scale All – execute all scaling actions
- Cloud Scale for Performance – only execute scaling actions that improve performance
- Cloud Scale for Savings – only execute scaling actions that reduce costs

The default action acceptance mode for these actions is *Manual*. When you examine the pending actions, only actions that satisfy the policy are allowed to execute. All other actions are read-only.

When policy conflicts arise, **Cloud Scale All** overrides the other two scaling options in most cases. For more information, see [Default and User-defined Automation Policies](#).

## Supported Instance Types for AWS VMs

Turbonomic considers all supported instance types when making scaling decisions for cloud VMs. If you want your VMs to *only scale to* or *avoid* certain instance types, create policies for those VMs.

You can view the instance types that Turbonomic supports from the user interface.

1. Navigate to **Settings > Policies**.
2. In the Policy Management page, search for and click **Virtual Machine Defaults**.
3. In the Configure Virtual Machine Policy page:
  - a. Scroll down to the bottom of the page.
  - b. Click **Add Scaling Constraint**.
  - c. Choose **Cloud Instance Types**.
  - d. Click **Edit**.

The policy page shows supported tiers for each cloud provider. A tier is a family of instance types, such as *M1* for GCP, *a1* for AWS and *Basic\_A1* for Azure. Expand a tier to see individual instance types and resource allocations.

## Scaling Prerequisites for AWS VMs

In AWS some instances require VMs to be configured in specific ways before they can scale to those instance types. If Turbonomic recommends scaling a VM that is not suitably configured onto one of these instances, then it sets the action to *Recommend*, and describes the reason. Turbonomic will not automate the action, even if you have set the action acceptance mode for that scope to *Automatic*. You can execute the action manually, after you have properly configured the instance.

Note that if you have VMs that you cannot configure to support these requirements, then you can set up a policy to keep Turbonomic from making these recommendations. Create a group that contains these VMs, and then create policy for that scope. In the policy, exclude instance types by configuring the Cloud Instance Types scaling constraint. For information about excluding instance types, see [Cloud Instance Types](#).

The instance requirements that Turbonomic recognizes are:

- Enhanced Network Adapters

Some VMs can run on instances that support Enhanced Networking via the Elastic Network Adapter (ENA), while others can run on instances that do not offer this support. Turbonomic can recommend scaling a VM that does not support ENA onto an instance that does. However, you must enable ENA on the VM before executing the scaling action. If you scale a non-ENA VM to an instance that requires ENA, then AWS cannot start up the VM after the scaling action.

For information about ENA configuration, visit this [page](#).

- Linux AMI Virtualization Type

An Amazon Linux AMI can use ParaVirtual (PV) or Hardware Virtual Machine (HVM) virtualization. Turbonomic can recommend scaling a PV VM to an HVM instance that does not include the necessary PV drivers.

To check the virtualization type of an instance, open the Amazon EC2 console to the Details pane, and review the Virtualization field for that instance.

- 64-bit vs 32-bit

Not all AWS instance can support a 32-bit VMs. Turbonomic can recommend scaling a 32-bit VM to an instance that only supports a 64-bit platform.

- NVMe Block

Some instances expose EBS volumes as NVMe block devices, but not all VMs are configured with NVMe drivers. Turbonomic can recommend scaling such a VM to an instance that supports NVMe. Before executing the action, you must install the NVMe drivers on the VM.

In addition, Turbonomic recognizes processor types that you currently use for your VM. For scale actions, Turbonomic keeps your VMs on instance types with compatible processors:

- GPU-based instances:

Turbonomic recognizes when your VM is on a GPU-based instance. To ensure the VM always stays on a compatible processor, Turbonomic does not recommend scale actions.

- ARM-based instances

If your VM is on an ARM-based instance, then Turbonomic will only recommend scaling to other compatible ARM-based instance types.

## Scaling Storage for AWS VMs

When a VM needs more storage capacity Turbonomic recommends actions to scale its volume to an instance that provides more storage. Note that AWS supports both Elastic Block Store (EBS) and Instance storage. Turbonomic recognizes these storage types as it recommends volume actions.

If the root storage for your VM is Instance Storage, then Turbonomic will not recommend an action. This is because Instance Storage is ephemeral, and such an action would cause the VM to lose all the stored data.

If the root storage is EBS, then Turbonomic recommends volume actions. EBS is persistent, and the data will remain after the action. However, if the VM uses Instance Storage for extra storage, then Turbonomic does not include that storage in its calculations or actions.

## Nodes in AWS EMR Clusters

Turbonomic treats nodes in AWS [EMR](#) clusters like regular VMs. As such, it could incorrectly generate scale actions for such nodes. After a node action executes, AWS detects the action as a defect, terminates the node, and replaces it with a new instance of the initial size. To avoid this issue, we recommend that you disable scale actions for nodes in EMR clusters.

AWS automatically assigns [system tags](#) to EMR clusters. To disable scale actions, create a VM group that uses these tags as a filter, and then create a VM policy that disables the 'Cloud Scale All' action type for the VM group.

## Actions for Azure VMs

Turbonomic can recommend the following actions:

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

See additional information for scale actions below.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing Azure reservations coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

If you need more capacity, then Turbonomic will recommend actions to purchase additional reservations.

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

- **Stop and Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

For details, see [Parking: Stop or Start Cloud Resources](#).

## Controlling Scale Actions for Azure VMs

For scale actions, you can create policies to control the scale actions that Turbonomic recommends. In those policies, choose from the following options:

- Cloud Scale All – execute all scaling actions
- Cloud Scale for Performance – only execute scaling actions that improve performance
- Cloud Scale for Savings – only execute scaling actions that reduce costs

The default action acceptance mode for these actions is *Manual*. When you examine the pending actions, only actions that satisfy the policy are allowed to execute. All other actions are read-only.

When policy conflicts arise, **Cloud Scale All** overrides the other two scaling options in most cases. For more information, see [Default and User-defined Automation Policies](#).

## Supported Instance Types for Azure VMs

Turbonomic considers all supported instance types when making scaling decisions for cloud VMs. If you want your VMs to *only scale to* or *avoid* certain instance types, create policies for those VMs.

You can view the instance types that Turbonomic supports from the user interface.

1. Navigate to **Settings > Policies**.
2. In the Policy Management page, search for and click **Virtual Machine Defaults**.
3. In the Configure Virtual Machine Policy page:
  - a. Scroll down to the bottom of the page.
  - b. Click **Add Scaling Constraint**.
  - c. Choose **Cloud Instance Types**.
  - d. Click **Edit**.

The policy page shows supported tiers for each cloud provider. A tier is a family of instance types, such as *M1* for GCP, *a1* for AWS and *Basic\_A1* for Azure. Expand a tier to see individual instance types and resource allocations.

## Azure Resource Group Discovery

To discover Azure Resource Groups, you can set up the following targets:

- Microsoft Azure service principle targets
- Microsoft Azure Enterprise Agreement (EA) targets

For Azure environments that include Resource Groups, Turbonomic discovers the Azure Resource Groups and the tags that are used to identify these groups.

In the Turbonomic user interface, to search for a specific Azure Resource Group, choose **Resource Groups** in the Search Page.

You can set the scope of your Turbonomic session to an Azure Resource Group by choosing a group in the Search results and clicking **Scope To Selection**.

You can also use Azure tags as filter criteria when you create a custom Turbonomic resource group. You can choose the Azure Resource Groups that match the tag criteria to be members of the new custom group.

To find the available tags for a specific Azure Resource Group, add the Basic Info chart configured with Related Tag Information to your view or custom dashboard. See [Basic Info Charts](#).

### Note:

When you inspect Resource Groups, Turbonomic does not currently show the billed costs for those Resource Groups.

## Azure Instance Requirements

In Azure environments, some instance types require workloads to be configured in specific ways, and some workload configurations require instance types that support specific features. When Turbonomic generates resize actions in Azure, these actions consider the following features:

- **Accelerated Networking (AN)**

In an Azure environment, not all instance types support AN, and not all workloads on AN instances actually enable AN. Turbonomic maintains a dynamic group of workloads that have AN enabled, and it assigns a policy to that group to exclude any templates that do not support AN. In this way, if a workload is on an instance that supports AN, and that workload has enabled AN, then Turbonomic will not recommend an action that would move the workload to a non-AN instance.

- **Azure Premium Storage**

Turbonomic recognizes whether a workload uses Premium Storage, and will not recommend a resize to an instance that does not support Azure Premium Storage.

In addition, Turbonomic recognizes processor types that you currently use for your workloads. If your workload is on a GPU-based instance, then Turbonomic will only recommend moves to other compatible GPU-based instance types. For these workloads, Turbonomic does not recommend resize actions.

## IOPS-aware Scaling for Azure VMs

Turbonomic considers IOPS utilization when making scaling decisions for Azure VMs. To measure utilization, Turbonomic takes into account a variety of attributes, such as per-disk IOPS utilization, whole VM IOPS utilization, cache settings, and IOPS capacity for the VMs. It also respects IOPS utilization and aggressiveness constraints that you set in VM policies. For details, see [Aggressiveness and Observation Periods](#).

Analysis impacts VM scaling decisions in different ways. For example:

- If your instance experiences IOPS bottlenecks, Turbonomic can recommend scaling up to a larger instance type to increase IOPS capacity, even if you do not fully use the current VCPU or VMEM resources.
- If your instance experiences underutilization of VMEM and VCPU, but high IOPS utilization, Turbonomic might not recommend scaling down. It might keep you on the larger instance to provide sufficient IOPS capacity.
- If the instance experiences underutilization of IOPS capacity along with normal utilization of other resources, you might see an action to resize to an instance that is very similar to the current one. If you inspect the action details, you should see that you are changing to a less expensive instance with less IOPS capacity.

## Actions for GCP VMs

Turbonomic can recommend the following actions:

- **Scale**

Change the VM instance to use a different instance type or tier to optimize performance and costs.

See additional information for scale actions below.

- **Discount-related actions**

If you have a high percentage of on-demand VMs, you can reduce your monthly costs by increasing Committed Use Discount (CUD) coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

Actions to purchase CUDs will be introduced in a future release.

- **Stop and Start** (also known as 'parking' actions)

Stop a VM for a given period of time to reduce your cloud expenses, and then start it at a later time.

For details, see [Parking: Stop or Start Cloud Resources](#).

- **Reconfigure**

GCP provides a specific set of machine types for each zone in a region. If you create a policy that restricts a VM to certain machine types and the zone it is currently on does not support all of those machine types, Turbonomic will recommend a reconfigure action as a way to notify you of the non-compliant VM.

For example, assume Zone A does not support machine types for the M1 family. When a VM in that zone applies a policy that restricts it to M1, Turbonomic will recommend that you reconfigure the VM.

## Points to Consider for GCP Scale Actions

- Turbonomic can generate scaling actions for the following VMs, but cannot execute the actions automatically:

- VMs with local SSDs

Turbonomic can recommend scaling to a machine type that supports local SSDs and the number of disks required by the VM, but blocks action execution due to prerequisite steps that you can only perform from GCP. You can view the prerequisite steps when you examine a pending action.

- VMs configured with a [minimum CPU platform](#)

GCP instance type families can support multiple CPU generations. A specific VM may be configured with a minimum CPU platform to prevent it from scaling to instance types with incompatible CPUs. When you examine a pending action for such a VM, verify that the recommended instance type runs a compatible CPU. Once verified, manually execute the action from GCP.

- **Note:**

Turbonomic does not recognize prioritized attributions you may have set for CUDs. For example, if you have prioritized all your CUD allotments for a single project, Turbonomic can still recommend actions to apply CUD to other projects in your environment.

- Since all GCP compute tiers have the same net throughput capacity, Turbonomic will not generate scaling actions in response to net throughput.
- Turbonomic does not recommend scaling actions for:

- [Spot VMs](#)

- **Note:**

Turbonomic discovers spot VMs, but does not recommend actions or monitor costs for these VMs.

- VMs running on [sole-tenant nodes](#)
- VMs with attached [GPUs](#)
- VMs in [managed instance groups](#)
- VMs running [custom machine types](#)

## Controlling Scale Actions for GCP VMs

For scale actions, you can create policies to control the scale actions that Turbonomic recommends. In those policies, choose from the following options:

- Cloud Scale All – execute all scaling actions
- Cloud Scale for Performance – only execute scaling actions that improve performance
- Cloud Scale for Savings – only execute scaling actions that reduce costs

The default action acceptance mode for these actions is *Manual*. When you examine the pending actions, only actions that satisfy the policy are allowed to execute. All other actions are read-only.

When policy conflicts arise, **Cloud Scale All** overrides the other two scaling options in most cases. For more information, see [Default and User-defined Automation Policies](#).

## Supported Instance Types for GCP VMs

Turbonomic considers all supported instance types when making scaling decisions for cloud VMs. If you want your VMs to *only scale to* or *avoid* certain instance types, create policies for those VMs.

You can view the instance types that Turbonomic supports from the user interface.

1. Navigate to **Settings > Policies**.
2. In the Policy Management page, search for and click **Virtual Machine Defaults**.
3. In the Configure Virtual Machine Policy page:
  - a. Scroll down to the bottom of the page.
  - b. Click **Add Scaling Constraint**.
  - c. Choose **Cloud Instance Types**.
  - d. Click **Edit**.

The policy page shows supported tiers for each cloud provider. A tier is a family of instance types, such as *M1* for GCP, *a1* for AWS and *Basic\_A1* for Azure. Expand a tier to see individual instance types and resource allocations.

## Cloud VM Uptime

For cloud VMs, Turbonomic includes *uptime* data in its cost calculations. This is especially important for VMs that do not run 24/7 and are charged on-demand rates. With uptime data, Turbonomic can calculate costs more accurately based on the amount of time a VM has been running.

The Action Details page shows uptime data for these VMs. Turbonomic calculates uptime based on the VM's age.





## Key Concepts

- Uptime

A percentage value that indicates how long a VM has been running over a period of time (age)

- Age

The number of days that a VM has existed since first discovery. For VMs older than 30 days, Turbonomic displays a value of **30+ days**, but only calculates uptime over the last 30 days.

For newly discovered VMs, age is 0 (zero) on the day of discovery. If the VM is running at the time of discovery, uptime is 100%. Otherwise, uptime is 0% and remains unchanged until the VM is powered on. Turbonomic recalculates uptime every hour and then refreshes the data shown in the user interface.

## Examples

- A VM that was first discovered 5 days (or 120 hours) ago and has been running for a total of 60 hours during that period has a current uptime value of 50%.
- A VM that was first discovered 2 months ago and has been running for a total of 180 hours over the last 30 days (or 720 hours) has a current uptime value of 25%.

## Cost Calculations Using Uptime Data

Turbonomic uses uptime data to calculate estimated on-demand costs for your cloud VMs. For details about calculations, see [Estimated On-demand Monthly Costs for Cloud VMs](#).

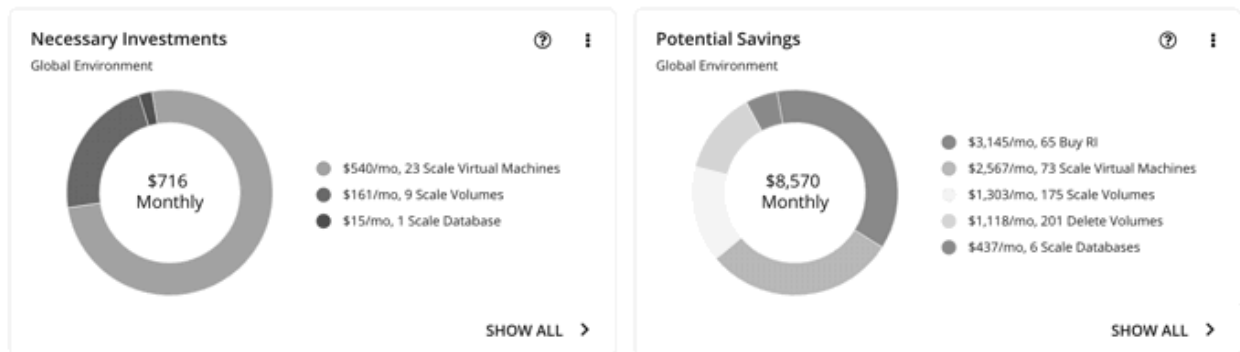
Uptime data impacts cost calculations, but not the actual scaling decisions that Turbonomic makes. These decisions rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

## Uptime Data in Charts

Turbonomic recalculates uptime data every hour and then updates the values shown in charts. The following charts reflect the cost impact of uptime-based calculations:

- Potential Savings and Necessary Investment charts

The projected amounts in these charts include on-demand costs for cloud VMs.

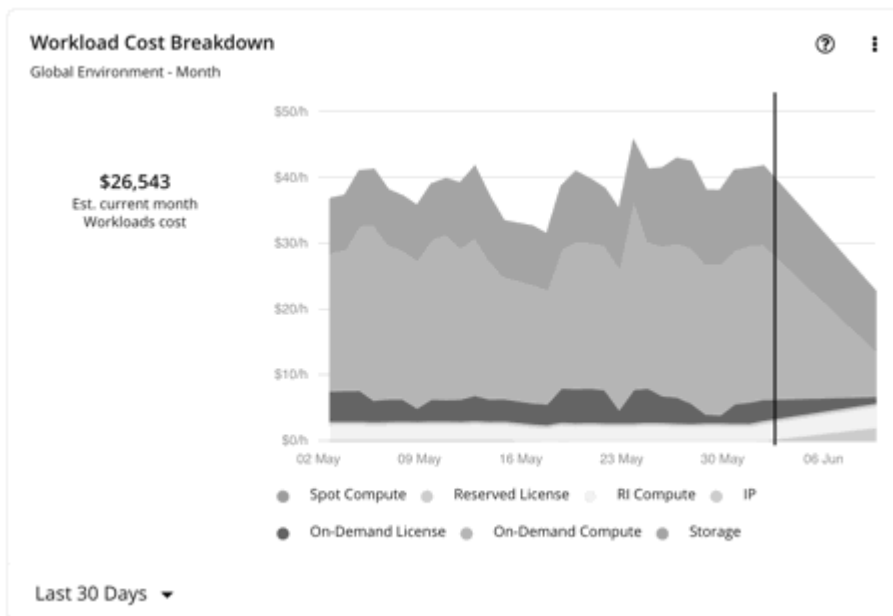


When you click **Show All** in these charts and view details for a pending VM action, the **Action Details** page shows on-demand costs before and after you execute the action, factoring in the VM's uptime value. The page also shows the VM's age.



- Workload Cost Breakdown chart

This chart shows estimated costs over time, including on-demand costs for VMs.



- The **Entity Information** chart shows the latest uptime and age data for a specific cloud VM.

|                        |                   |
|------------------------|-------------------|
| Number of VCPUs        | 4                 |
| Region                 | azure-Canada East |
| Account                | [REDACTED]        |
| Resource Group         | [REDACTED]        |
| Uptime ⓘ               | 86.9%             |
| Last Modification Time | N/A               |
| Attachment State       | Attached          |
| Vendor ID [EA - PT2]   | [REDACTED]        |
| Age ⓘ                  | 30+ days          |

## Estimated On-demand Costs for Cloud VMs

Turbonomic considers a variety of factors when calculating *Estimated On-demand Monthly Cost* for a cloud VM.

| VIRTUAL MACHINE DETAILS |               |                 |                               |                |            |               |            |  |  |
|-------------------------|---------------|-----------------|-------------------------------|----------------|------------|---------------|------------|--|--|
| NAME                    | ID            | AGE             | ACCOUNT                       | REGION         |            |               |            |  |  |
| PT_Consistent_S...      | [REDACTED]    | 30+ days        | [REDACTED]                    | aws-EU (Paris) |            |               |            |  |  |
| VCPU PERCENTILE         |               | VMEM PERCENTILE |                               | NET THROUGHPUT |            | IO THROUGHPUT |            |  |  |
| 1 %                     | 0.6 % ↓       | 94 %            | 47 % ↓                        | 0 %            | 0 %        | 0 %           | 0 %        |  |  |
| 800 MHz                 | 1.4 GHz       | 1 GiB           | 2 GiB                         | 468.8 MB/s     | 468.8 MB/s | 260.6 MB/s    | 260.6 MB/s |  |  |
| ON-DEMAND RATE ⓘ        | RI COVERAGE ⓘ | UPTIME ⓘ        | EST. ON-DEMAND MONTHLY COST ⓘ |                |            |               |            |  |  |
| \$0.012/h               | 50% → 0%      | 95.3%           | \$4.1/mo → \$15/mo            |                |            |               |            |  |  |

## AWS VMs and Azure VMs Without License Costs

### Cost Calculation

For these VMs, the calculation for Estimated On-demand Monthly Cost can be expressed as follows:

$$\text{On-demand Rate} * \text{Usage Not Covered by Discounts} * \text{Uptime} * 730 = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Rate** is the hourly cost for a VM's instance type *without* discount coverage (AWS RIs/ Savings Plans or Azure reservations).

- For AWS, this rate includes all license costs, but not storage or IP. You can obtain on-demand rates via Amazon EC2 On-demand Pricing.
- For Azure, the rate does *not* include license costs, storage, or IP. You can obtain on-demand rates via [Azure Pricing Calculator](#).

**Note:**

Azure VMs covered by Azure Hybrid Benefit do not have license costs.

- **Usage Not Covered by Discounts** is the percentage of hourly VM usage not covered by any discount. For example:

- Discount Coverage = 20% (0.2)
- Usage Not Covered by Discounts = 80% (0.8)

- **Uptime** is a percentage value that indicates how long a VM has been running since it was first discovered by Turbonomic. For VMs discovered more than 30 days ago, Turbonomic only calculates uptime over the last 30 days.

To estimate monthly on-demand costs, Turbonomic projects the current uptime value into the future. It assumes that future uptime will be similar to the current uptime.

- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.

The listed items above impact cost calculations, but not the actual scaling decisions that Turbonomic makes. These decisions rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

**Example**

Assume the following data for a pending scale action for an AWS VM:

The screenshot shows a 'VIRTUAL MACHINE DETAILS' panel with the following data:

| NAME               | ID         | AGE      | ACCOUNT    | REGION         |
|--------------------|------------|----------|------------|----------------|
| PT_Consistent_S... | [Redacted] | 30+ days | [Redacted] | aws-EU (Paris) |

| VCPU PERCENTILE                  | VMEM PERCENTILE              | NET THROUGHPUT                     | IO THROUGHPUT                      |
|----------------------------------|------------------------------|------------------------------------|------------------------------------|
| 1% → 0.6% ↓<br>800 MHz → 1.4 GHz | 94% → 47% ↓<br>1 GiB → 2 GiB | 0% → 0%<br>468.8 MB/s → 468.8 MB/s | 0% → 0%<br>260.6 MB/s → 260.6 MB/s |

| ON-DEMAND RATE        | RI COVERAGE | UPTIME | EST. ON-DEMAND MONTHLY COST |
|-----------------------|-------------|--------|-----------------------------|
| \$0.012/h → \$0.021/h | 50% → 0%    | 95.3%  | \$4.1/mo → \$15/mo          |

|                                                                                  | Current Values | Values After Action Execution |
|----------------------------------------------------------------------------------|----------------|-------------------------------|
| On-demand Rate                                                                   | \$0.012/hr     | \$0.021/hr                    |
| Discount Coverage                                                                | 50% (0.5)      | 0% (0.0)                      |
| Usage Not Covered by Discounts<br><i>(calculated based on discount coverage)</i> | 50% (0.5)      | 100% (1.0)                    |
| Uptime                                                                           | 95.3% (.953)   |                               |

Turbonomic calculates the following:

| VIRTUAL MACHINE DETAILS |           |                 |          |                               |
|-------------------------|-----------|-----------------|----------|-------------------------------|
| NAME                    | ID        | AGE             | ACCOUNT  | REGION                        |
| PT_Consistent_S...      |           | 30+ days        |          | aws-EU (Paris)                |
| VCPU PERCENTILE         |           | VMEM PERCENTILE |          | NET THROUGHPUT                |
| 1 %                     | 0.6 % ↓   | 94 %            | 47 % ↓   | 0 %                           |
| 800 MHz                 | 1.4 GHz   | 1 GiB           | 2 GiB    | 468.8 MB/s                    |
| ON-DEMAND RATE ⓘ        |           | RI COVERAGE ⓘ   | UPTIME ⓘ | EST. ON-DEMAND MONTHLY COST ⓘ |
| \$0.012/h               | \$0.021/h | 50% → 0%        | 95.3%    | \$4.1/mo → \$15/mo            |

- *Current* Estimated On-demand Monthly Cost:

$$0.012 * 0.5 * 0.953 * 730 = 4.1$$

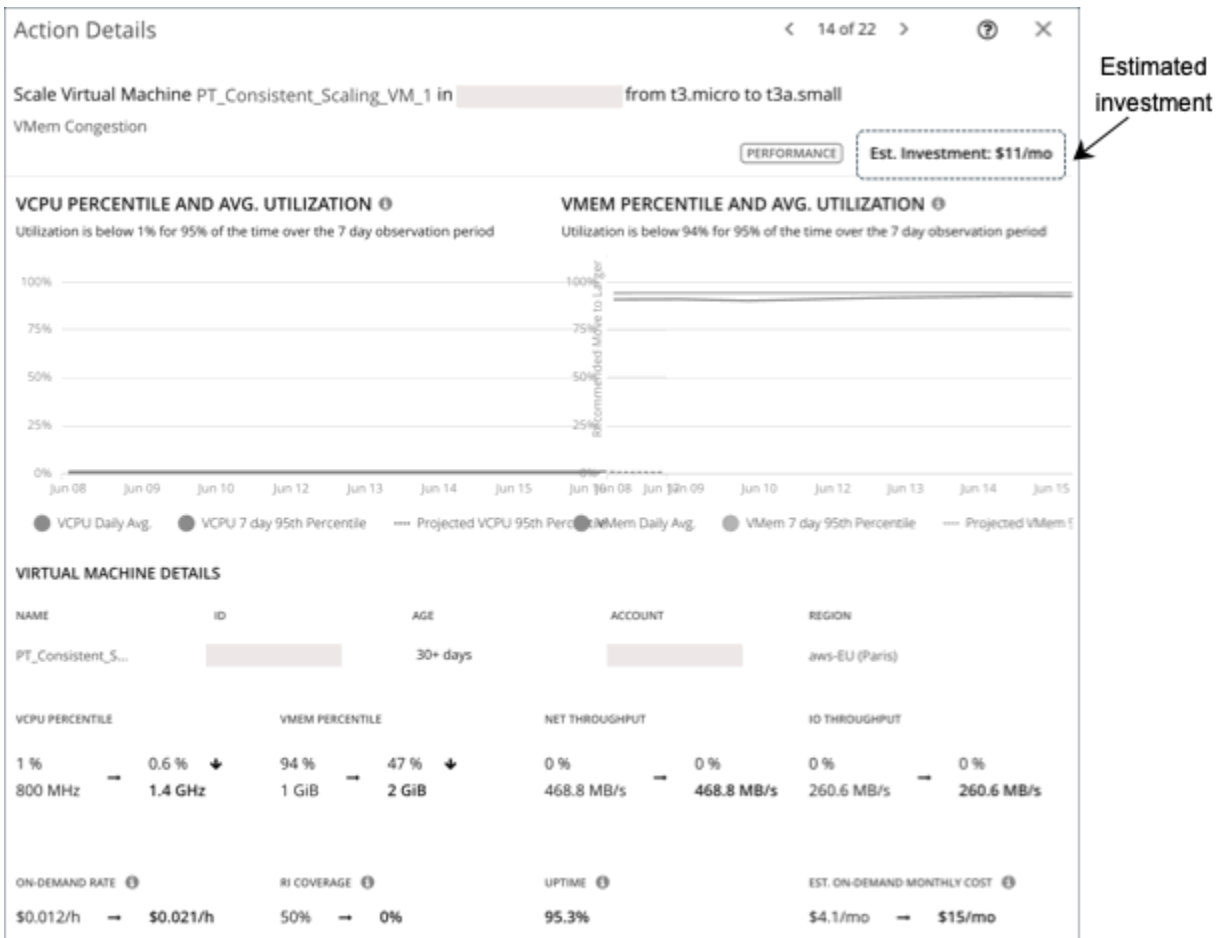
- Estimated On-demand Monthly Cost *after* executing the action:

$$0.021 * 1.0 * 0.953 * 730 = 15$$

**Note:**

Turbonomic rounds the calculated values that it displays in the user interface.

Since the Estimated On-demand Monthly Cost is projected to increase from \$4.1/month to \$15/month, Turbonomic treats the action as an investment and shows an estimated investment of \$11/month.



## Azure VMs with License Costs

### Cost Calculation

For VMs with license costs, Turbonomic first calculates the *On-demand Compute Rate*, which it then uses to calculate *Estimated On-demand Monthly Costs*.

#### 1. On-demand Compute Rate Calculation

The calculation for On-demand Compute Rate can be expressed as follows:

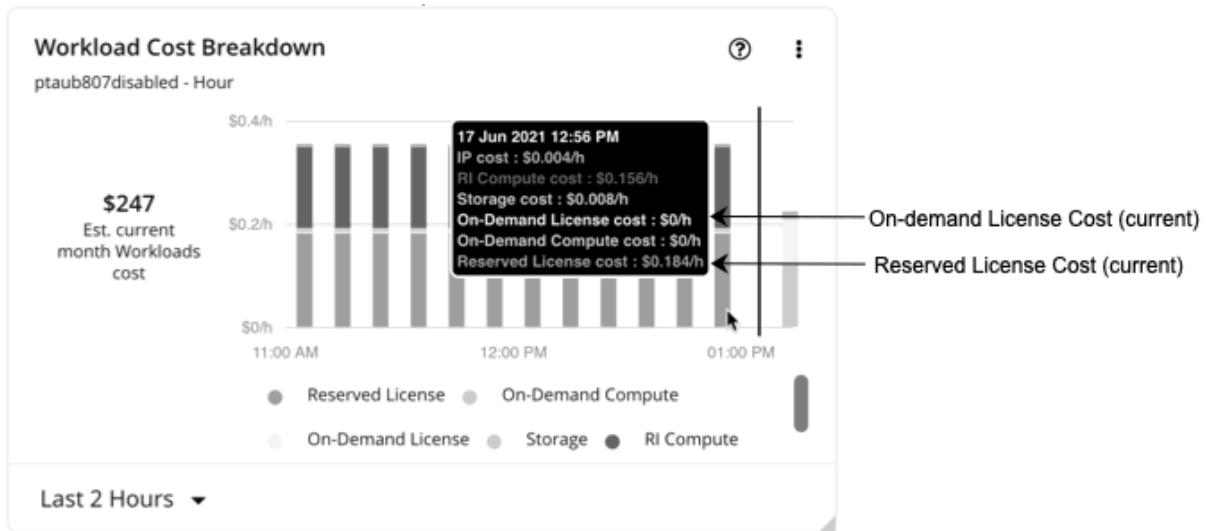
$$\text{On-demand Rate} - (\text{Reserved License Cost} + \text{On-demand License Cost}) = \text{On-demand Compute Rate}$$

Where:

- **On-demand Rate** is the hourly cost for a VM's instance type *without* reservation coverage. This does *not* include license costs, storage, or IP. You can obtain on-demand rates via [Azure Pricing Calculator](#).
- **Reserved License Cost** and **On-demand License Cost** are the hourly costs for the VM's licenses. You can obtain license costs via Azure Pricing Calculator or the Turbonomic user interface.

From the user interface, set the scope to the Azure VM and then see the Workload Cost Breakdown chart. In the chart, set the time frame to Last 2 Hours, and then:

- Hover over the second to the last bar in the chart to obtain the *current* On-demand License Cost and Reserved License Cost.



- Hover over the last bar (after the vertical line) in the chart to obtain the On-demand License Cost and Reserved License Cost *after* you execute actions.



The *On-demand Compute Rate* and *License Cost (On-demand and Reserved)* are then used to calculate Estimated On-demand Monthly Costs.

## 2. Estimated On-demand Monthly Cost Calculation

The calculation can be expressed as follows:

$$(\text{On-demand Compute Rate} * \text{Usage Not Covered by Reservations}) + \text{License Cost} * \text{Uptime} * 730 = \text{Estimated On-demand Monthly Cost}$$

Where:

- **Usage Not Covered by Reservations** is the percentage of hourly VM usage not covered by any reservation. For example:
  - Reservation Coverage = 20% (0.2)
  - Usage Not Covered by Reservations = 80% (0.8)
- **License Cost** is the sum of On-demand License Cost and Reserved License Cost.
- **Uptime** is a percentage value that indicates how long a VM has been running since it was first discovered by Turbonomic. For VMs discovered more than 30 days ago, Turbonomic only calculates uptime over the last 30 days



To estimate monthly on-demand costs, Turbonomic projects the current uptime value into the future. It assumes that future uptime will be similar to the current uptime.

- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.

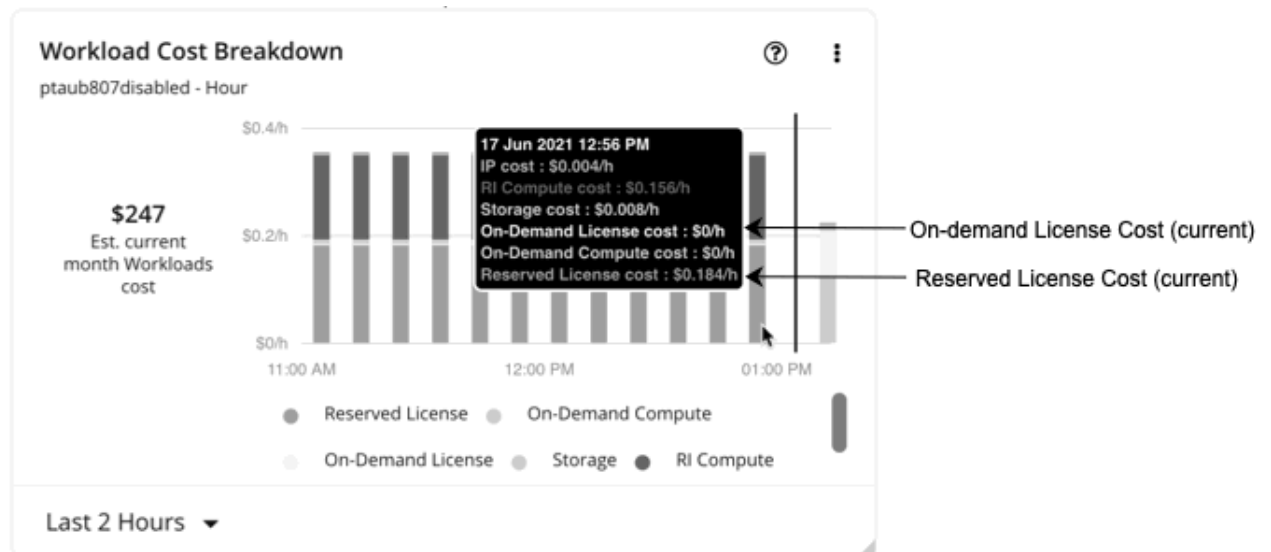
The listed items above impact cost calculations, but not the actual scaling decisions that Turbonomic makes. These decisions rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

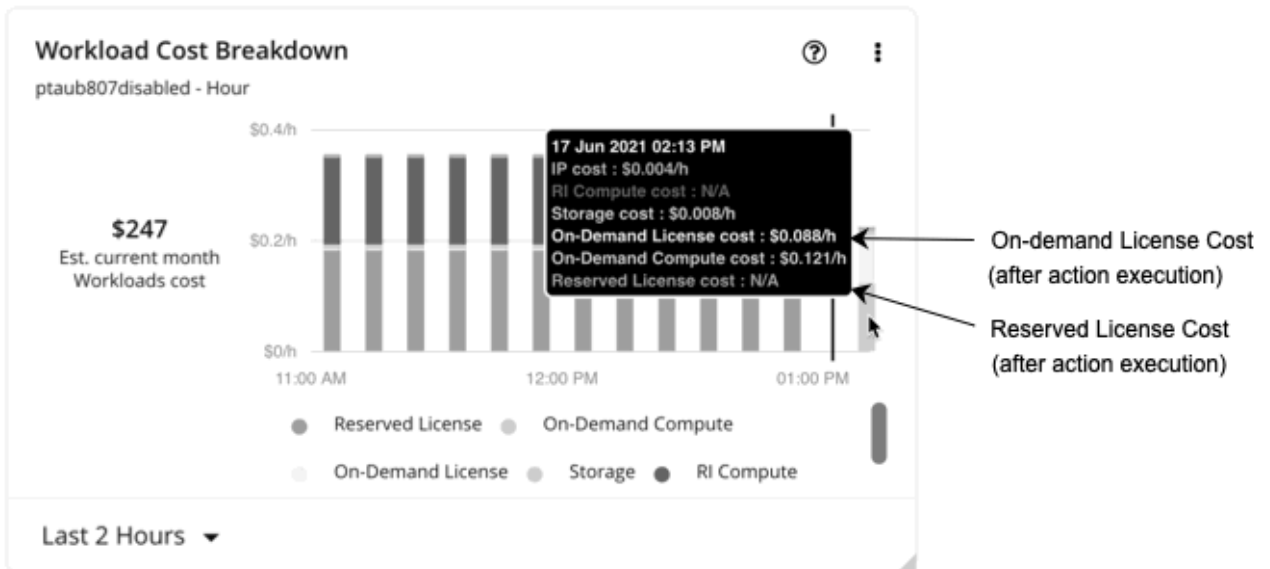
### Example

Assume the following data for a pending scale action for an Azure VM with license costs:

| VIRTUAL MACHINE DETAILS          |                    |                 |                  |                               |
|----------------------------------|--------------------|-----------------|------------------|-------------------------------|
| NAME                             | ID                 | AGE             | SUBSCRIPTION     | LOCATION                      |
| ptaub807disable...               | 25b9c0ce-0e08-4... | 30+ days        | EA - Development | azure-East US 2               |
| VCPUs PERCENTILE                 |                    | VMEM PERCENTILE |                  | VM IOPS PERCENTILE            |
| 2 %                              | 2.9 % ↑            | 9 %             | 18 % ↑           | 0 %                           |
| 33.2 GHz                         | 22.7 GHz           | 32 GiB          | 16 GiB           | 6,400 IOPS                    |
| VM STORAGE THROUGHPUT PERCENTILE |                    |                 |                  |                               |
| 0 %                              |                    | 0 %             |                  |                               |
| 96 MB/s                          |                    | 48 MB/s         |                  |                               |
| ON-DEMAND RATE ⓘ                 |                    | RI COVERAGE ⓘ   | UPTIME ⓘ         | EST. ON-DEMAND MONTHLY COST ⓘ |
| \$0.45/h → \$0.218/h             |                    | 100% → 0%       | 96.1%            | \$129/mo → \$153/mo           |

|                | Current Values | Values After Action Execution |
|----------------|----------------|-------------------------------|
| On-demand Rate | \$0.45/hr      | \$0.218/hr                    |





|                        | Current Values | Values After Action Execution |
|------------------------|----------------|-------------------------------|
| On-demand License Cost | \$0/hr         | \$0.088/hr                    |
| Reserved License Cost  | \$0.184/hr     | N/A                           |

1. Turbonomic first calculates the following:

- *Current* On-demand Compute Rate:

$$0.45 - (0.184 + 0) = 0.266$$

- On-demand Compute Rate *after* executing the action:

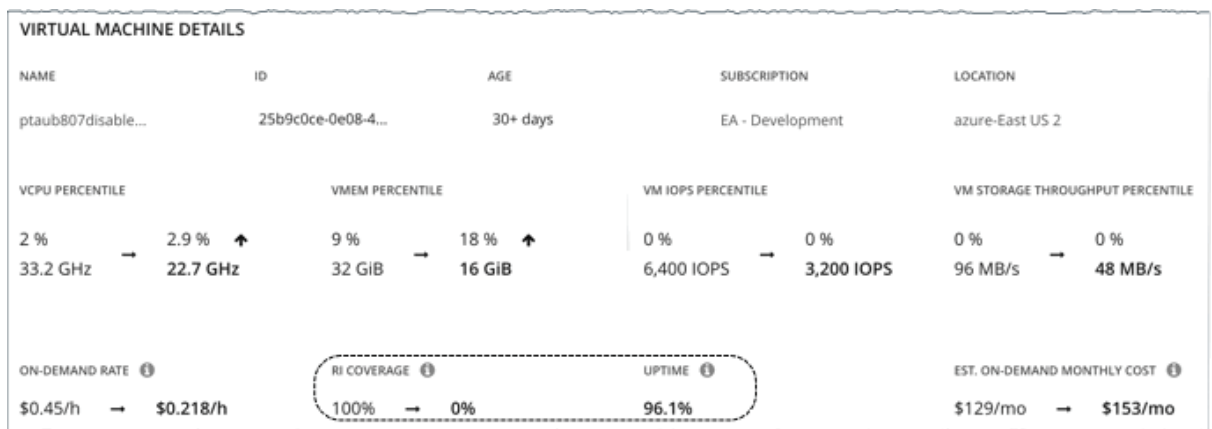
$$0.218 - (0 + 0.088) = 0.13$$

2. Turbonomic can now calculate Estimated On-demand Monthly Cost based on:

- On-demand Compute Rate

|                        | Current Values | Values After Action Execution |
|------------------------|----------------|-------------------------------|
| On-demand Compute Rate | \$0.266/hr     | \$0.13/hr                     |

- Usage Not Covered by Reservations and Uptime



|                                                                                       | Current Values | Values After Action Execution |
|---------------------------------------------------------------------------------------|----------------|-------------------------------|
| Reservation Coverage                                                                  | 100% (1.0)     | 0% (0.0)                      |
| Usage Not Covered by Reservations ( <i>calculated based on reservation coverage</i> ) | 0% (0.0)       | 100% (1.0)                    |
| Uptime                                                                                | 96.1% (.961)   |                               |

Turbonomic calculates the following:

**VIRTUAL MACHINE DETAILS**

|                    |                    |          |                  |                 |
|--------------------|--------------------|----------|------------------|-----------------|
| NAME               | ID                 | AGE      | SUBSCRIPTION     | LOCATION        |
| ptaub807disable... | 25b9c0ce-0e08-4... | 30+ days | EA - Development | azure-East US 2 |

|                                      |                                 |                                      |                                  |
|--------------------------------------|---------------------------------|--------------------------------------|----------------------------------|
| VCPU PERCENTILE                      | VMEM PERCENTILE                 | VM IOPS PERCENTILE                   | VM STORAGE THROUGHPUT PERCENTILE |
| 2 % → 2.9 % ↑<br>33.2 GHz → 22.7 GHz | 9 % → 18 % ↑<br>32 GiB → 16 GiB | 0 % → 0 %<br>6,400 IOPS → 3,200 IOPS | 0 % → 0 %<br>96 MB/s → 48 MB/s   |

|                                          |                            |                   |                                                      |
|------------------------------------------|----------------------------|-------------------|------------------------------------------------------|
| ON-DEMAND RATE ⓘ<br>\$0.45/h → \$0.218/h | RI COVERAGE ⓘ<br>100% → 0% | UPTIME ⓘ<br>96.1% | EST. ON-DEMAND MONTHLY COST ⓘ<br>\$129/mo → \$153/mo |
|------------------------------------------|----------------------------|-------------------|------------------------------------------------------|

- *Current* Estimated On-demand Monthly Cost:

$$(0.266 * 0.0) + 0.184 * 0.961 * 730 = 129$$

- Estimated On-demand Monthly Cost *after* executing the action:

$$(0.13 * 1.0) + 0.088 * 0.961 * 730 = 153$$

**Note:**

Turbonomic rounds the calculated values that it displays in the user interface.

Since the on-demand cost is projected to increase from \$129/month to \$153/month, Turbonomic treats the action as an investment and shows an estimated investment of \$24/month.



Estimated investment

## Cloud VM Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Note:

The policy settings discussed in this topic only apply to cloud scale actions. For details about settings for parking actions (stop and start), see [Parking: Stop or Start Cloud Resources](#).






## Action Automation and Orchestration

For details about cloud VM actions, see [Cloud VM Actions](#).

### Cloud Scale

| Action                      | Default Mode | AWS, Azure, and GCP |
|-----------------------------|--------------|---------------------|
| Cloud Scale All             | Manual       | <b>Auto</b>         |
| Cloud Scale for Performance | Manual       | <b>Auto</b>         |
| Cloud Scale for Savings     | Manual       | <b>Auto</b>         |

### Other Actions

| Action                         | Default Mode | AWS and Azure                                                                     | GCP                                                                                 |
|--------------------------------|--------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Buy discounts                  | Recommend    |  | Not yet supported                                                                   |
| Provision Kubernetes node (VM) | Manual       |  |  |
| Suspend Kubernetes node (VM)   | Manual       |  |  |

## Scaling Target Utilization

### For VCPU, VMEM, and IO/Net Throughput Utilization:

These advanced settings determine how much you would like a scope of workloads to utilize their resources. These are fixed settings that override the way Turbonomic calculates the optimal utilization of resources. You should only change these settings after consulting with Technical Support.

While these settings offer a way to modify how Turbonomic recommends actions, in most cases you should never need to use them. If you want to control how Turbonomic recommends actions to resize workloads, you can set the aggressiveness per the percentile of utilization, and set the length of the sample period for more or less elasticity on the cloud.

| Attribute                                 | Default Value                                                                                       |
|-------------------------------------------|-----------------------------------------------------------------------------------------------------|
| Scaling Target VCPU Utilization           | 70<br>The target utilization as a percentage of VCPU capacity.                                      |
| Scaling Target VMEM Utilization           | 90<br>The target utilization as a percentage of memory capacity.                                    |
| Scaling Target IO Throughput Utilization  | 70<br>The target utilization as a percentage of IO throughput (Read and Write) capacity.            |
| Scaling Target Net Throughput Utilization | 70<br>The target utilization as a percentage of network throughput (Inbound and Outbound) capacity. |

### For IOPS Utilization:

Turbonomic uses this setting in conjunction with aggressiveness constraints to control scaling actions for VMs. You can set the aggressiveness per the percentile of utilization, and set the length of the sample period for more or less elasticity on the cloud.

| Attribute                                        | Default Value                                                                               |
|--------------------------------------------------|---------------------------------------------------------------------------------------------|
| Scaling Target IOPS Utilization (Azure VMs only) | 70<br>For Azure environments, the target percentile value Turbonomic will attempt to match. |

For details on how IOPS utilization affects scaling decisions, see [IOPS-aware Scaling for Azure VMs](#).

## Aggressiveness and Observation Periods

Turbonomic uses these settings to calculate utilization percentiles for vCPU, vMEM, and IOPS (Azure VMs only). It then recommends actions to improve utilization based on the observed values for a given time period.

### • Aggressiveness

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 95th percentile. The percentile utilization is the highest value that 95% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce the capacity for CPU on a VM. Without using a percentile, Turbonomic never resizes below the recognized peak utilization. For most VMs, there are moments when peak CPU reaches high levels, such as during reboots, patching, and other maintenance tasks. Assume utilization for a VM peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce allocated CPU for that VM.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single CPU burst to 100%, but for 95% of the samples CPU never exceeded 50%. If you set **Aggressiveness** to 95th Percentile, then Turbonomic can see this as an opportunity to reduce CPU allocation for the VM.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 100th and 99th Percentile – More performance. Recommended for critical workloads that need maximum guaranteed performance at all times, or workloads that need to tolerate sudden and previously unseen spikes in utilization, even though sustained utilization is low.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings. This assures application performance while avoiding reactive peak sizing due to transient spikes, thus allowing you to take advantage of the elastic ability of the cloud.
- 90th Percentile – More efficiency. Recommended for non-production workloads that can stand higher resource utilization.

By default, Turbonomic uses samples from the last 30 days. Use the **Max Observation Period** setting to adjust the number of days. To ensure that there are enough samples to analyze and drive scaling actions, set the **Min Observation Period**.

### • Max Observation Period

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 30 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the database has fewer days' data then it uses all of the stored historical data.

You can make the following settings:

- Less Elastic – Last 90 Days
- Recommended – Last 30 Days
- More Elastic – Last 7 Days

Turbonomic recommends an observation period of 30 days following the monthly workload maintenance cycle seen in many organizations. VMs typically peak during the maintenance window as patching and other maintenance tasks are carried out. A 30-day observation period means that Turbonomic can capture these peaks and increase the accuracy of its sizing recommendations.

You can set the value to 7 days if workloads need to resize more often in response to performance changes. For workloads that cannot handle changes very often or have longer usage periods, you can set the value to 90 days.

- **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

- More Elastic – None
- Less Elastic – 7 Days

## Cloud Instance Types

| Attribute            | Default Value |
|----------------------|---------------|
| Cloud Instance Types | None          |

By default, Turbonomic considers all instance types currently available for scaling when making scaling decisions for VMs. However, you may have set up your cloud VMs to *only scale to* or *avoid* certain instance types to reduce complexity and cost, improve discount utilization, or meet application demand. Use this setting to identify the instance types that VMs can scale to.

**Note:**

Under most circumstances, when a cloud provider offers a new instance type that is meant to replace an older type, the provider offers it at a lower cost. However, a provider may provide a new instance type with identical costs as the older instance types. If this occurs, and capacity and cost are equal, Turbonomic cannot ensure that it chooses the newer instance type. To work around this issue, you can create an Action Automation policy that excludes the older instance type.

Click **Edit** to set your preferences. In the new page that displays, expand a **cloud tier** (a family of instance types, such as *a1* for AWS or *B-series* for Azure) to see individual instance types and the resources allocated to them. If you have several cloud providers, each provider will have its own tab.

Select your preferred instance types or cloud tiers, or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

If you selected a cloud tier and the service provider deploys new instance types to that tier later, then those instance types will automatically be included in your policy. Be sure to review your policies

periodically to see if new instance types have been added to a tier. If you do not want to scale to those instance types, update the affected policies.

## Consistent Resizing

| Attribute                  | Default Setting |
|----------------------------|-----------------|
| Enable Consistent Resizing | Off             |

### Consistent Resizing for User-defined Automation Policies

When you create a policy for a group of VMs and turn on Consistent Resizing, Turbonomic resizes all the group members to the same size, such that they all support the top utilization of each resource commodity in the group. For example, assume VM A shows top utilization of CPU, and VM B shows top utilization of memory. A resize action would result in all the VMs with CPU capacity to satisfy VM A, and memory capacity to satisfy VM B.

For an affected resize, the Actions List shows individual resize actions for each of the VMs in the group. If you automate resizes, Turbonomic executes each resize individually in a way that avoids disruption to your workloads.

Use this setting to enforce the same template across all VMs in a group when resizing VMs on the public cloud. In this way, Turbonomic can enforce a rule to size all the VMs in a group equally.

### Consistent Resizing for Auto-discovered Groups

In public cloud environments, Turbonomic discovers groups that should keep all their VMs on the same template, and then creates read-only policies for them to implement Consistent Resizing. The details of this discovery and the associated policy vary depending on the cloud provider.

- Azure

Turbonomic discovers Azure availability sets and scale sets.

- For availability sets, Turbonomic does *not* enable Consistent Resizing, but it can recommend scale actions for individual VMs in the availability set.

When a scale action for a VM in an availability set fails due to insufficient resources in the compute cluster, the action remains pending. When you hover over the pending action, you will see a message indicating that action execution has been temporarily disabled due to a previous execution error in the availability set. Turbonomic assumes that all other VMs in the availability set will fail to scale due to the same resource issue, so it creates a temporary policy that disables action execution for the availability set. Specifically, this policy sets the action acceptance mode for scale actions to *Recommend* and stays in effect for 730 hours (one month). This means that for the duration of the policy, Turbonomic will continue to generate read-only, non-executable scale actions for individual VMs, so you can evaluate their resource requirements and plan accordingly. You can delete this policy if you need to re-enable action execution in the availability set.

- For scale sets, Turbonomic automatically enables Consistent Resizing across all the VMs in the group. You can choose to execute all the actions for such a group, either manually or automatically. In that case, Turbonomic executes the resizes one VM at a time. If you do not need to resize all the members of a given scale set to a consistent template, create another policy for that scope and turn off Consistent Resizing.

- AWS

Turbonomic discovers Auto Scaling Groups and automatically enables Consistent Resizing across all the VMs in each group. You can choose to execute all the actions for such a group, either manually or automatically. In that case, Turbonomic executes the resizes one VM at a time. If you do not need to resize all the members of a given Auto Scaling Group to a consistent template, create another policy for that scope and turn off Consistent Resizing.



If you select one or all actions for the group either manually or automatically, Turbonomic will change the Launch Configuration for the Auto Scaling Group but it will not terminate the EC2 instances.

Below are some use cases for employing Consistent Resizing for a group.

- If you have deployed load balancing for a group, then all the VMs in the group should experience similar utilization. In that case, if one VM needs to be resized, then it makes sense to resize them all consistently.
- A common HA configuration on the public cloud is to deploy mirror VMs to different availability zones, where the given application runs on only one of the VMs at a given time. The other VMs are on standby to recover in failover events. Without Consistent Resizing, Turbonomic would tend to size down or suspend the unused VMs, which would make them unready for the failover situation.

When working with Consistent Resizing, consider these points:

- You should not mix VMs in a group that has a Consistent Resizing policy, with other groups that enable Consistent Resizing. One VM can be a member of more than one group. If one VM (or more) in a group with Consistent Resizing is also in another group that has Consistent Resizing, then both groups enforce Consistent Resizing together, for all their group members.
- If one VM (or more) is in a group with Consistent Resizing turned on, and the same VMs are in a group with Consistent Resizing turned off, the affected VMs assume the ON setting. This is true if you created both groups, or if Turbonomic created one of the groups for Azure Scale Sets or AWS Auto Scaling Groups.
- For any group of VMs that enables Consistent Resizing, you should not mix the associated target technologies. For example, one group should not include VMs that are managed on both Azure and AWS platforms.
- Charts that show actions and risks assign the same risk statement to all the affected VMs. This can seem confusing. For example, assume one VM needs to resize to address vCPU risk, and 9 other VMs are set to resize consistently with it. Then charts will state that 10 VMs need to resize to address vCPU risks.

## Ignore NVMe Constraints

For AWS, Turbonomic recognizes when a VM instance includes an NVMe driver. To respect NVMe constraints, it will not recommend scaling to an instance type that does not also include an NVMe driver. If you ignore NVMe constraints, then Turbonomic is free to scale the instance to a type that does not include an NVMe driver.

| Attribute                             | Default Setting |
|---------------------------------------|-----------------|
| Ignore NVMe Constraints<br>(AWS only) | Off             |

## Instance Store Aware Scaling

| Attribute                    | Default Setting |
|------------------------------|-----------------|
| Instance Store Aware Scaling | Off             |

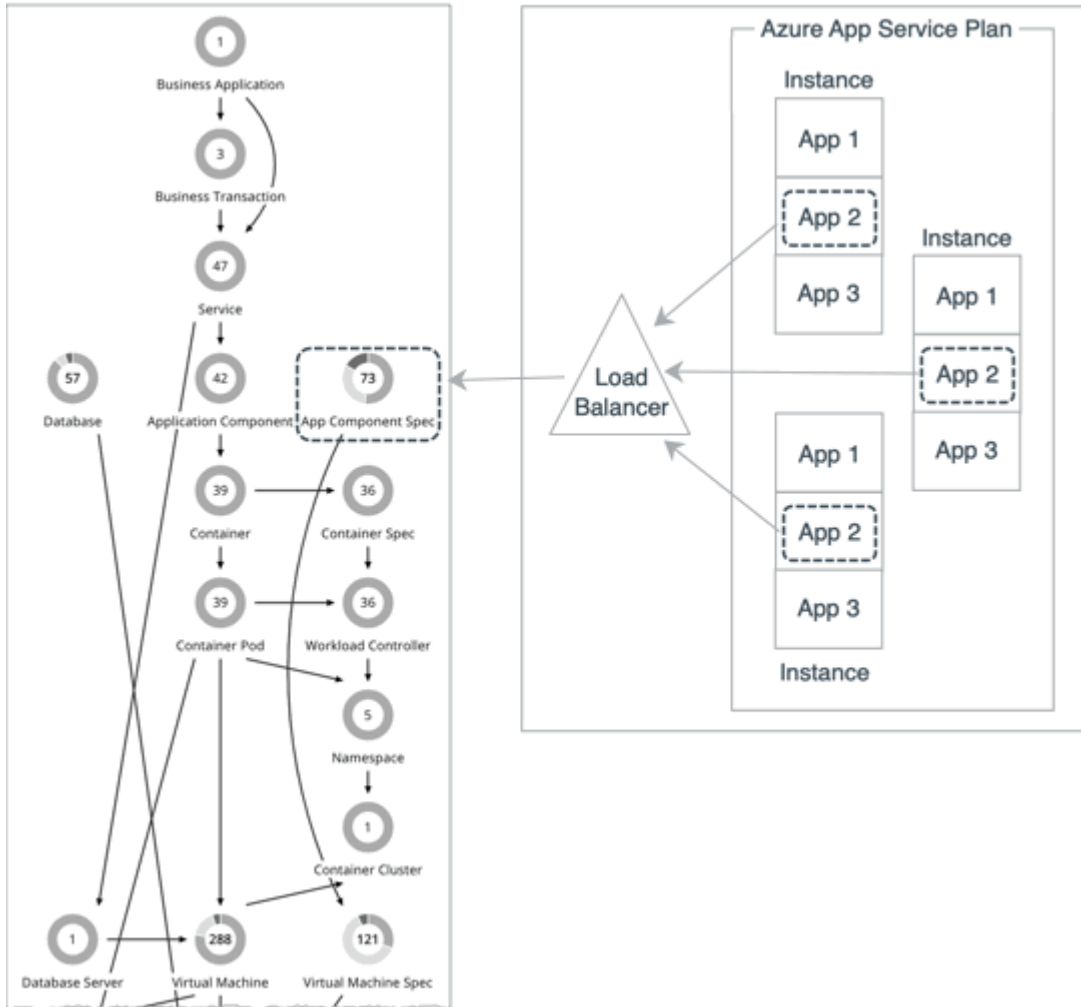
The template for your workload determines whether the workload can use an *instance store*, and it determines the instance store capacity. As Turbonomic calculates a resize or move action, it can recommend a new template that does not support instance stores, or that does not provide the same instance store capacity.

To ensure that resize actions respect the instance store requirements for your workloads, turn on **Instance Store Aware Scaling** for a given VM or for a group of VMs. When you turn this on for a given

scope of VMs, then as it calculates move and resize actions, Turbonomic will only consider templates that support instance stores. In addition, it will not move a workload to a template that provides less instance store capacity.

## App Component Spec

In [Azure App Service](#) deployments, an App Component Spec represents a set of app instances comprising a single web application. Turbonomic discovers App Component Specs when you add an Azure target with the necessary permissions.



### Note:

For a list of permissions, see [Azure Permissions](#).

Turbonomic also discovers the *plans* that provide resources to app instances. The supply chain shows these plans as [Virtual Machine Specs](#) and links them with App Component Specs to establish their relationship.

## Synopsis

| Synopsis  |                                  |
|-----------|----------------------------------|
| Provides: | App services to end users        |
| Consumes: | Resources from App Service plans |

## Synopsis

|                     |               |
|---------------------|---------------|
| Discovered through: | Azure targets |
|---------------------|---------------|

## Monitored Resources

Turbonomic monitors the following resources:

- Response Time

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

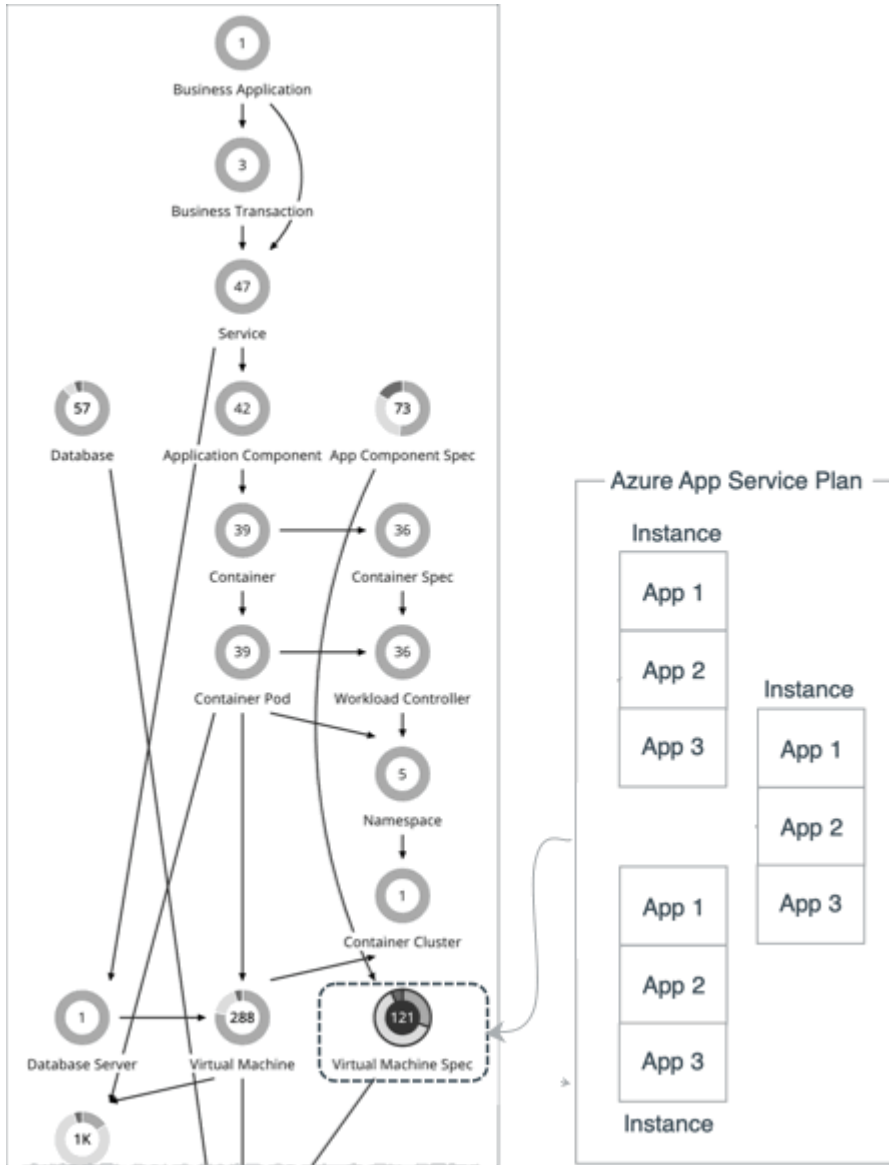
## Actions

### None

Turbonomic does not recommend actions for App Component Specs, but it does recommend actions for the underlying Virtual Machine Specs. For details, see [Virtual Machine Spec Actions](#).

## Virtual Machine Spec

In [Azure App Service](#), *plans* define CPU, memory, and storage resources that are available to VM instances to run apps. When you add an Azure target with the necessary permissions, Turbonomic discovers the plans associated with apps, and shows them as Virtual Machine Specs in the supply chain. Currently, Turbonomic discovers all plans, except App Service Environment v3 I4, I5, and I6.



**Note:**

For a list of permissions, see [Azure Permissions](#).

Points to consider:

- Azure App Service offers several types of apps, including web apps, mobile apps, API apps, and logic apps. Turbonomic discovers the plans associated with these apps, but only recommends scale actions for plans associated with web apps. If a plan is no longer associated with any type of app, Turbonomic will recommend that you delete it.
- For web apps, Turbonomic also discovers the app instances that consume resources from a plan, and shows them as **App Component Specs** in the supply chain. The supply chain links **App Component Specs** with **Virtual Machine Specs** to establish their relationship.
- VM instances underlying a plan scale as a group. For this reason, Turbonomic represents these VM instances as a single **Virtual Machine Spec** entity and does *not* monitor them individually. The Entity Information chart for a **Virtual Machine Spec** shows the current number of VM instances, while resource charts (such as the **Virtual CPU** and **Virtual Memory** charts) show aggregated metrics for all VM instances.

## Synopsis

| Synopsis            |                                             |
|---------------------|---------------------------------------------|
| Provides:           | Resources to apps (via App Component Specs) |
| Consumes:           | Resources from Azure regions                |
| Discovered through: | Azure targets                               |

## Monitored Resources

Turbonomic monitors the following resources:

- **Virtual Memory (VMem)**  
Virtual Memory is the measurement of memory utilized by the entity.
- **Virtual CPU (VCPU)**  
Virtual CPU is the measurement of CPU utilized by the entity.
- **Storage Amount**  
Storage Amount is the measurement of storage capacity utilized by the entity.
- **Number of Replicas**  
Number of Replicas is the total number of VM instances underlying an App Service plan.

## Actions

Turbonomic can recommend the following actions:

- **Scale**  
Scale Azure App Service plans to optimize app performance or reduce costs, while complying with business policies.
- **Delete**  
Delete empty Azure App Service plans as a cost-saving measure. A plan is considered empty if it is not hosting any running apps.

## Scale Actions

Turbonomic supports vertical scaling actions for provisioned App Service plans. These actions change the *size* of all VM instances underlying a plan (for example, from small to large, or large to medium). Horizontal scaling actions, which change the *number* of VM instances underlying a plan, are currently not supported.

Vertical scaling recommendations rely on a variety of factors, including:

- [Resource utilization percentiles](#)
- [On-demand monthly costs](#)
- VM instance count  
Turbonomic will only recommend vertical scaling actions on plans with six or less VM instances.
- Scaling eligibility
  - Eligible for scaling – Basic, Standard, Premium v2, Premium v3, Isolated, Isolated v2
  - Not eligible for scaling – Workflow Standard, Elastic Premium, Free, Shared, Dynamic/Serverless
- Azure-enforced constraints, including:

- Region – Only recommend instance types in regions where they are available
- Server rack – Only recommend instance types on server racks where they are available
- Zone redundancy – If zone redundancy is enabled, only recommend instance types that support zone redundancy
- Deployment slots – Only recommend instance types that support the currently configured number of deployment slots that can be added to apps
- Hybrid connections – Only recommend scaling to instance types that support the currently configured number of hybrid connections for a plan

**Note:**

To see the number of deployment slots and hybrid connections configured for a plan, set the scope to the corresponding Virtual Machine Spec and then view the Entity Information chart.

- Scaling constraints that you set in Turbonomic [policies](#) for Virtual Machine Specs

For example, you can set a constraint if you want App Service plans to *only scale to or avoid* certain instance types.

For scale actions, you can create policies to control the scale actions that Turbonomic recommends. In those policies, choose from the following options:

- Cloud Scale All – execute all scaling actions
- Cloud Scale for Performance – only execute scaling actions that improve performance
- Cloud Scale for Savings – only execute scaling actions that reduce costs

The default action acceptance mode for these actions is *Manual*. When you examine the pending actions, only actions that satisfy the policy are allowed to execute. All other actions are read-only.

When policy conflicts arise, **Cloud Scale All** overrides the other two scaling options in most cases. For more information, see [Default and User-defined Automation Policies](#).

## Delete Actions

When Turbonomic discovers an empty plan (i.e., a plan that is not hosting any running apps), it will immediately recommend that you delete the plan as a cost-saving measure. Turbonomic can recommend deleting provisioned App Service plans, as well as Elastic Premium and Workflow Standard plans.

If a currently empty plan is not deleted and is subsequently discovered as used, Turbonomic removes the delete action attached to it.

Delete actions include the 'Days Empty' information that indicates how long a plan has been empty.

**Action Details** < 10 of 52 > ? X

Delete Empty P2v2 App Service Plan paas-asp-zone-redundant from [redacted] ↓ \$147.46/mo SAVINGS

Increase savings

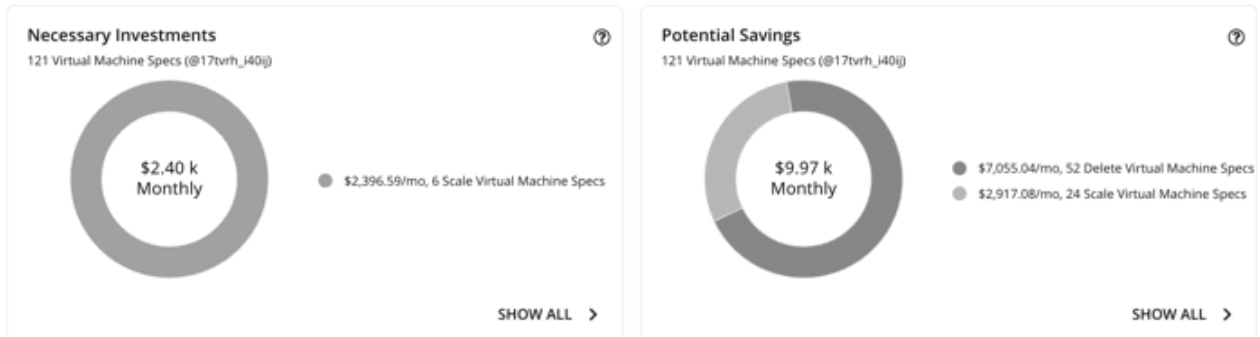
| ACTION ESSENTIALS      |                                                    | APP SERVICE PLAN DETAILS |                            |
|------------------------|----------------------------------------------------|--------------------------|----------------------------|
| State                  | 📄 Action can be accepted and executed immediately. | Name                     | paas-asp-zone-redundant 🗑️ |
| Non-Disruptive         | ✓ Downtime is not required to execute.             | Id                       | paas-asp-zone-redundant 🗑️ |
| Reversible             | ✗ Action cannot be manually reverted.              | Subscription             | [redacted] 🗑️              |
| <b>RESOURCE IMPACT</b> |                                                    | Location                 | azure-East US 🗑️           |
|                        | <b>CURRENT</b>                                     | <b>AFTER ACTIONS</b>     | <b>Days Empty</b> 7        |
| Virtual Memory         | 7 GB                                               | -                        |                            |
| Virtual CPU            | 18.36 GHz                                          | -                        |                            |

You can control the delete actions that Turbonomic recommends, based on the 'Days Empty' value that you set. For example, if you want Turbonomic to only generate delete actions for plans that have been empty for at least 5 days, perform these steps:

1. In the default policy for Virtual Machine Specs, *disable* delete actions.
2. Create a dynamic group of Virtual Machine Specs and set the 'Days Empty' filter to Days Empty > = 5.
3. Create a custom Virtual Machine Spec policy, set the scope to the group that you just created, and then *enable* delete actions in that policy.

## Actions in Charts

Use the Necessary Investments and Potential Savings charts to view pending Virtual Machine Spec actions. These charts show total monthly investments and savings, assuming you execute all the actions.



Click **Show All** for each chart to review and execute the actions.

The table shows the following:

- Actions that are pending for each Virtual Machine Spec
- Savings or investments for each Virtual Machine Spec

## Utilization Charts for Scale Actions

Turbonomic uses percentile calculations to measure resource utilization, and drive scaling actions that improve overall utilization and reduce costs. When you examine the details for a pending scaling action on an App Service plan, you will see charts that highlight resource *utilization percentiles* for a given observation period, and the projected percentiles after you execute the action.

Action Details 1 of 24

Scale Virtual Machine Spec ASP-CloudPaaS-3e from I2v2 to I1v2 in ↓ \$845.34/mo SAVINGS

Underutilized vCPU, vMEM

**VCPU PERCENTILE AND AVG. UTILIZATION**

vCPU Utilization is below 2% for 95% of the time over the 30 day observation period

**VMEM PERCENTILE AND AVG. UTILIZATION**

vMEM Utilization is below 14% for 95% of the time over the 30 day observation period

**STORAGE AMOUNT AVG. UTILIZATION**

Storage Amount Utilization average is equal to 0%

**ACTION ESSENTIALS**

- State: Action can be accepted and executed immediately.
- Non-Disruptive: Downtime is required to execute.
- Reversible: Action can be manually reverted.

**APP SERVICE PLAN DETAILS**

- Name: ASP-CloudPaaS-3e
- Id: ASP-CloudPaaS-3e
- Subscription: [Redacted]
- Location: azure-East US

|                                 | CURRENT  | AFTER ACTIONS |             |
|---------------------------------|----------|---------------|-------------|
| Plan Tier                       | I2v2     | I1v2          |             |
| VMem, Capacity                  | 16 GB    | 8 GB          | ↓ 8 GB      |
| VMem, P95th Utilization         | 14%      | 28%           | ↑ 14 %      |
| VCPU, Capacity                  | 34.1 GHz | 17.05 GHz     | ↓ 17.05 GHz |
| VCPU, P95th Utilization         | 2%       | 4%            | ↑ 2 %       |
| Storage, Capacity               | 0.98 TB  | 0.98 TB       | -           |
| Storage, Utilization            | 0%       | 0%            | -           |
| Number Of Replicas, Capacity    | 100      | 100           | -           |
| Number Of Replicas, Utilization | 3%       | 3%            | -           |

|                | CURRENT       | AFTER ACTIONS |               |
|----------------|---------------|---------------|---------------|
| Compute Cost   | \$563.56/mo   | \$281.78/mo   | ↓ \$281.78/mo |
| Instance Count | 3             | 3             |               |
| Total Cost     | \$1,690.68/mo | \$845.34/mo   | ↓ \$845.34/mo |
| Total Savings  |               | \$845.34/mo   |               |

**POLICIES**

The charts also plot *daily average utilization* for your reference. If you have previously executed scaling actions on the App Service plan, you can see the resulting improvements in daily average utilization. Put together, these charts allow you to easily recognize utilization trends that drive Turbonomic's scaling recommendations.

**Note:**

You can set scaling constraints in Virtual Machine Spec policies to refine the percentile calculations. For details, see [Scaling Sensitivity](#).

**Disruptiveness and Reversibility of Scale Actions**

Turbonomic always recommends scaling to a different instance type, so all scaling actions are disruptive and require downtime. You can reverse an action by scaling an App Service plan back to its original instance type.

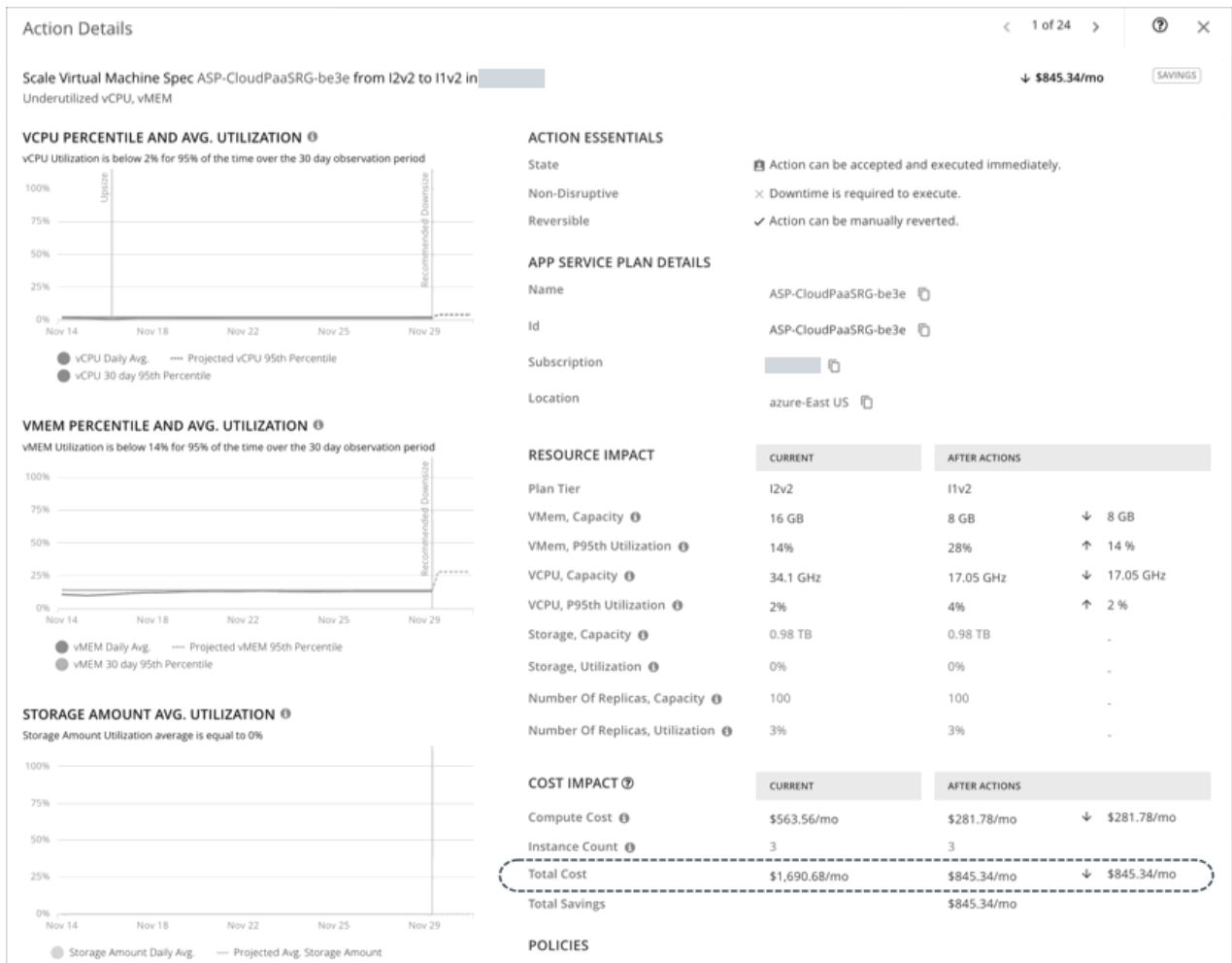
**Estimated On-demand Monthly Costs for Azure App Service Plans**

Turbonomic considers a variety of factors when calculating estimated on-demand monthly costs for Azure App Service plans.

**Note:**

Azure App Service plans appear as Virtual Machine Spec entities in the supply chain.





## Cost Calculation

The calculation for estimated on-demand monthly cost can be expressed as follows:

$$(\text{On-demand Compute Rate} \times 730) \times \text{Number of Instances} = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Compute Rate** is the **hourly** cost for an App Service plan's instance type. You can obtain on-demand rates via [App Service Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to calculate monthly costs.
- **Number of Instances** is the total number of VM instances underlying the App Service plan.

The listed items above impact cost calculations and the scaling decisions that Turbonomic makes. These decisions also rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

## Example

Assume the following data for a pending action to scale an Azure Service plan from the I2V2 to the I1V1 instance type.

|                        | Current Values | Values After Action Execution |
|------------------------|----------------|-------------------------------|
| On-demand Compute Rate | \$0.772/hr     | \$0.386/hr                    |

|                     | Current Values | Values After Action Execution |
|---------------------|----------------|-------------------------------|
| Number of Instances | 3              | 3                             |

Turbonomic calculates the following:

- *Current* estimated on-demand monthly cost:

$$(\$0.772 * 730) * 3 = \$1690.68/\text{Mo.}$$

- Estimated on-demand monthly cost *after* executing the action:

$$(\$0.386 * 730) * 3 = \$845.34/\text{Mo.}$$

**Note:**

Turbonomic rounds the calculated values that it displays in the user interface.

The estimated on-demand monthly cost is projected to decrease from \$1690.68/month to \$845.34/month, as shown in the Details section of the pending action.

| COST IMPACT ⓘ    | CURRENT       | AFTER ACTIONS |               |
|------------------|---------------|---------------|---------------|
| Compute Cost ⓘ   | \$563.56/mo   | \$281.78/mo   | ↓ \$281.78/mo |
| Instance Count ⓘ | 3             | 3             |               |
| Total Cost       | \$1,690.68/mo | \$845.34/mo   | ↓ \$845.34/mo |
| Total Savings    |               | \$845.34/mo   |               |
| POLICIES         |               |               |               |

Turbonomic treats the action as a cost-saving measure, and shows total savings of \$845.34/month.

| COST IMPACT ⓘ    | CURRENT       | AFTER ACTIONS |               |
|------------------|---------------|---------------|---------------|
| Compute Cost ⓘ   | \$563.56/mo   | \$281.78/mo   | ↓ \$281.78/mo |
| Instance Count ⓘ | 3             | 3             |               |
| Total Cost       | \$1,690.68/mo | \$845.34/mo   | ↓ \$845.34/mo |
| Total Savings    |               | \$845.34/mo   |               |
| POLICIES         |               |               |               |

### Estimated On-demand Monthly Savings for Empty Azure App Service Plans

Turbonomic considers an empty App Service plan's on-demand compute rate and VM instance count when calculating the estimated on-demand monthly savings that you would realize when you delete the plan. A plan is considered empty if it is not hosting any running apps.

**Note:**

Azure App Service plans appear as Virtual Machine Spec entities in the supply chain.

Action Details
< 10 of 52 >
?
×

**Delete Empty P2v2 App Service Plan paas-asp-zone-redundant from**  ↓ \$147.46/mo SAVINGS

Increase savings

|                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>ACTION ESSENTIALS</b></p> <p>State <span style="font-size: 0.8em;">📄 Action can be accepted and executed immediately.</span></p> <p>Non-Disruptive <span style="font-size: 0.8em;">✔ Downtime is not required to execute.</span></p> <p>Reversible <span style="font-size: 0.8em;">✗ Action cannot be manually reverted.</span></p> | <p><b>APP SERVICE PLAN DETAILS</b></p> <p>Name <span style="float: right;">paas-asp-zone-redundant <span style="font-size: 0.8em;">📄</span></span></p> <p>Id <span style="float: right;">paas-asp-zone-redundant <span style="font-size: 0.8em;">📄</span></span></p> <p>Subscription <span style="float: right;"><span style="background-color: #ccc; padding: 2px 10px;"></span> <span style="font-size: 0.8em;">📄</span></span></p> <p>Location <span style="float: right;">azure-East US <span style="font-size: 0.8em;">📄</span></span></p> <p>Days Empty <span style="float: right;">7</span></p> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                        |                |                      |  |
|------------------------|----------------|----------------------|--|
| <b>RESOURCE IMPACT</b> | <b>CURRENT</b> | <b>AFTER ACTIONS</b> |  |
| Virtual Memory         | 7 GB           | -                    |  |
| Virtual CPU            | 18.36 GHz      | -                    |  |

|                                                             |                |                      |               |
|-------------------------------------------------------------|----------------|----------------------|---------------|
| <b>COST IMPACT</b> <span style="font-size: 0.8em;">?</span> | <b>CURRENT</b> | <b>AFTER ACTIONS</b> |               |
| Compute Cost <span style="font-size: 0.8em;">📄</span>       | \$147.46/mo    | N/A                  |               |
| Instance Count <span style="font-size: 0.8em;">📄</span>     | 1              | N/A                  |               |
| Total Cost                                                  | \$147.46/mo    | \$0.00/mo            | ↓ \$147.46/mo |
| Total Savings                                               |                | \$147.46/mo          |               |

## Savings Calculation

The calculation for estimated on-demand monthly savings can be expressed as follows:

$(\text{On-demand Compute Rate} \times 730) \times \text{Number of Instances} = \text{Estimated On-demand Monthly Savings}$

Where:

- **On-demand Compute Rate** is the **hourly** cost for an App Service plan's instance type. You can obtain on-demand rates via [App Service Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to calculate monthly savings.
- **Number of Instances** is the total number of VM instances underlying the App Service plan.

## Example

Assume the following data for a pending action to delete an empty Azure Service plan on the P2V2 instance type.

|                        | Current Values |
|------------------------|----------------|
| On-demand Compute Rate | \$0.202/hr     |
| Number of Instances    | 1              |

Turbonomic calculates savings as follows:

$$(\$0.202 \times 730) \times 1 = \$147.46/\text{Mo.}$$

### Note:

Turbonomic rounds the calculated values that it displays in the user interface.

Turbonomic shows total savings of \$147.46/month.

| COST IMPACT <sup>?</sup>    | CURRENT                   | AFTER ACTIONS |               |
|-----------------------------|---------------------------|---------------|---------------|
|                             | Compute Cost <sup>!</sup> | \$147.46/mo   | N/A           |
| Instance Count <sup>!</sup> | 1                         | N/A           |               |
| Total Cost                  | \$147.46/mo               | \$0.00/mo     | ↓ \$147.46/mo |
| Total Savings               |                           | \$147.46/mo   |               |

## Virtual Machine Spec Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about Virtual Machine Spec actions, see [Virtual Machine Spec Actions](#).

| Action                      | Default Mode |
|-----------------------------|--------------|
| Cloud Scale All             | Manual       |
| Cloud Scale for Performance | Manual       |
| Cloud Scale for Savings     | Manual       |
| Delete Virtual Machine Spec | Manual       |

## Scaling Sensitivity

Turbonomic uses a percentile of utilization over the specified observation period. This gives sustained utilization and ignores short-lived bursts.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

Turbonomic uses these settings to calculate utilization percentiles for vCPU and vMEM.

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 95th percentile. The percentile utilization is the highest value that 95% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce vCPU capacity. Without using a percentile, Turbonomic never scales below the recognized peak utilization. Assume utilization peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce resources for the Virtual Machine Spec.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single burst to 100%, but for 95% of the samples,

utilization never exceeded 50%. If you set **Aggressiveness** to 95th Percentile, then Turbonomic can see this as an opportunity to reduce resource allocation.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 99th or 100th Percentile – More performance. Recommended for critical Virtual Machine Specs that need maximum guaranteed performance at all times, or those that need to tolerate sudden and previously unseen spikes in utilization, even though sustained utilization is low.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings. This assures performance while avoiding reactive peak sizing due to transient spikes, thus allowing you to take advantage of the elastic ability of the cloud.
- 90th Percentile – More efficiency. Recommended for Virtual Machine Specs that can stand higher resource utilization.

By default, Turbonomic uses samples from the last 14 days. Use the **Max Observation Period** setting to adjust the number of days.

• **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 14 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the volume has fewer days' data then it uses all of the stored historical data.

Choose from the following settings:

- Less Elastic – Last 30 Days
- Recommended – Last 14 Days
- More Elastic – Last 3 or 7 Days

• **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

Choose from the following settings:

- More Elastic – None
- Less Elastic – 1, 3, or 7 Days

## Cloud Instance Types

| Attribute            | Default Value |
|----------------------|---------------|
| Cloud Instance Types | None          |

By default, Turbonomic considers all instance types currently available for scaling when making scaling decisions for Virtual Machine Specs. However, you may have set up your Virtual Machine Specs to *only scale to* or *avoid* certain instance types to reduce complexity and cost, improve discount utilization, or meet application demand. Use this setting to identify the instance types that Virtual Machine Specs can scale to.

Click **Edit** to set your preferences. In the new page that displays, expand a **cloud tier** (a family of instance types, such as *Basic*) to see individual instance types and the resources allocated to them.

Select your preferred instance types or cloud tiers, or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

If you selected a cloud tier and the service provider deploys new instance types to that tier later, then those instance types will automatically be included in your policy. Be sure to review your policies periodically to see if new instance types have been added to a tier. If you do not want to scale to those instance types, update the affected policies.

## Scaling Target Utilization

The utilization that you set here specifies the percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute | Default Value |
|-----------|---------------|
| VCPU      | 70            |
| VMEM      | 90            |
| Storage   | 90            |

These advanced settings determine how much you would like a scope of workloads to utilize their resources. These are fixed settings that override the way Turbonomic calculates the optimal utilization of resources. You should only change these settings after consulting with Technical Support.

While these settings offer a way to modify how Turbonomic recommends actions, in most cases you should never need to use them. If you want to control how Turbonomic recommends actions to resize workloads, you can set the aggressiveness per the percentile of utilization, and set the length of the sample period for more or less elasticity on the cloud.

## Database Server (Cloud)

In AWS public cloud environments, a Database Server is a relational database that you have configured using AWS Relational Database Service (RDS). Turbonomic discovers RDS instances through your AWS targets, and then generates scaling actions as needed.

### Note:

Azure SQL Databases discovered by Turbonomic appear as *Database* entities in the supply chain. For details, see [Database \(Cloud\)](#).

## Synopsis



## Synopsis

|                     |                                                        |
|---------------------|--------------------------------------------------------|
| Budget:             | A cloud Database Server has unlimited budget.          |
| Provides:           | Database services to cloud applications and end users  |
| Consumes:           | Compute and storage resources in the availability zone |
| Discovered through: | AWS targets                                            |

## Permissions

Turbonomic requires the following permissions for AWS RDS Database Servers:

- Monitoring permissions:
  - cloudwatch:GetMetricData
  - pi:GetResourceMetrics
  - rds:DescribeDBInstances
  - rds:DescribeDBParameters
  - rds:ListTagsForResource
  - rds:DescribeOrderableDBInstanceOptions
- Action execution permissions:
  - rds:ModifyDBInstance
  - rds:StartDBInstance
  - rds:StopDBInstance

For the full list of permissions, see [AWS Permissions](#).

## Monitored Resources

Turbonomic monitors the following resources:

- Virtual Memory (VMem)

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU (VCPU)

Virtual CPU is the measurement of CPU utilized by the entity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

- DB Cache Hit Rate

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

- Connection

Connection is the measurement of database connections utilized by applications.

## Actions

- **Stop** and **Start** (also known as 'parking' actions)

Stop a Database Server for a given period of time to reduce your cloud expenses, and then start it at a later time.

- **Scale**

Scale compute and storage resources to optimize performance and costs.

## Stop and Start Actions for Database Servers

Turbonomic supports 'parking' actions for cloud resources. These actions *stop* your cloud resources for a given period of time to help you reduce your cloud expenses, and then *start* these resources later when you need them. You can enforce parking actions on demand or according to a schedule.

For details, see [Parking: Stop or Start Cloud Resources](#).

## Scale Actions for Database Servers

To recommend accurate scaling actions, Turbonomic analyzes resource utilization percentiles and collects relevant metrics (such as connections utilization) from AWS. It also takes into consideration constraints defined in [policies](#).

Consider the following scenarios and actions:

- To address vCPU congestion, Turbonomic can recommend scaling a Database Server to the instance type that can adequately meet demand at the lowest possible cost. If vCPU is underutilized, it can recommend scaling to a smaller instance type.
- To address IOPS congestion, Turbonomic can recommend increasing provisioned IOPS or scaling the Database Server to a different storage type. For gp2 storage, it can recommend increasing disk size to increase provisioned IOPS. After executing these actions, Turbonomic will not recommend new actions for the next six hours, in compliance with AWS's "cooldown" period for EBS storage.
- Turbonomic analyzes DB cache hit rate before making vMem scaling decisions. To perform its analysis, it collects cache hit rate metrics for Database Servers with [Performance Insights](#) enabled.

For Database Servers with cache hit rate metrics, Turbonomic considers at least 90% cache hit rate to be optimal. This percentage value is not configurable.

- A cache hit rate value equal to or greater than 90% indicates efficiency. For this reason, Turbonomic will not recommend an action even if vMem utilization is high. If vMem utilization is low, it will recommend scaling to a smaller instance type.



- When the cache hit rate is below 90%, Turbonomic will also not recommend an action, provided that vMem utilization remains low. If vMem utilization is high, then it will recommend scaling to a larger instance type.

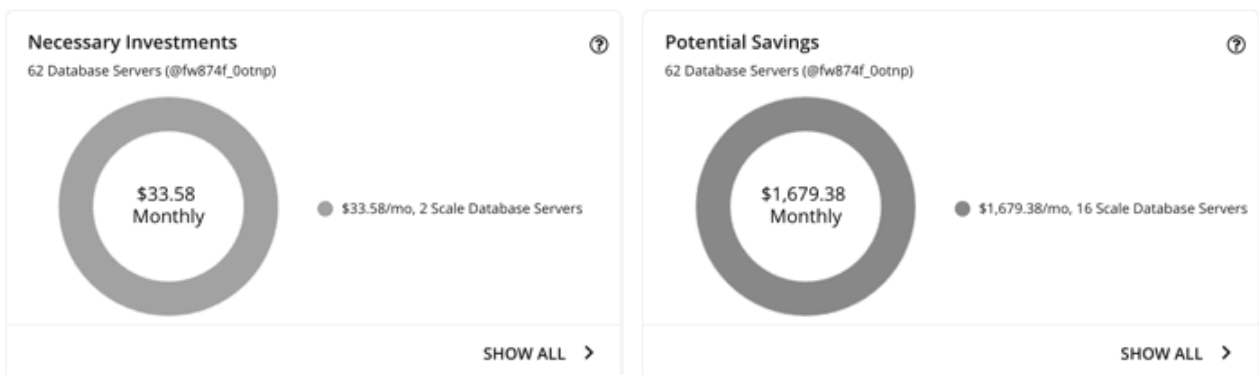
Notes on Performance Insights and cache hit rate metrics:

- Performance Insights is enabled by default on a majority of AWS Database Servers. In the Turbonomic user interface, you can use Search and then apply the Performance Insights filter to see which Database Servers have Performance Insights enabled.
- If Performance Insights is disabled or is not supported for your AWS Database Server engines or regions, Turbonomic will not have cache hit rate metrics to analyze and will therefore not generate actions in direct response to vMem utilization. For a list of supported engines and regions, see this [AWS page](#).
- An action to scale to a different instance type in response to vCPU utilization might also include vMem changes, but vMem utilization alone (without cache hit rate metrics) will not drive actions.

Turbonomic also considers Connections utilization and capacity when making scaling decisions. It collects utilization metrics from CloudWatch and calculates capacity based on the maximum number of simultaneous connections configured for the Database Server. The maximum number varies by Database Server engine type and memory allocation, and is set in the [parameter group](#) associated with a Database Server. Turbonomic currently supports Database Servers associated with parameter groups that use default values. For example, consider a MySQL Database Server that is on a `db.t3.large` instance type with 8 GB (8589934592 bytes) of memory, and is associated with a parameter group that uses the default value `{DBInstanceClassMemory/12582880}`. In this case, Turbonomic calculates capacity as 682 connections (or  $\{8589934592/12582880\}$ ). When Turbonomic generates an action due to vMem underutilization and sees that Connections utilization is only 15% of capacity (or roughly 100 connections), it picks a smaller instance type that is adequate for both the vMem and Connections requirements of the Database Server. For example, it could pick `db.t2.small`, which provides 2 GB of memory and a maximum of 170 connections.

## Scale Actions in Charts

Use the Necessary Investments and Potential Savings charts to view pending Database Server actions. These charts show total monthly investments and savings, assuming you execute all the actions.



Click **Show All** for each chart to review and execute the actions.

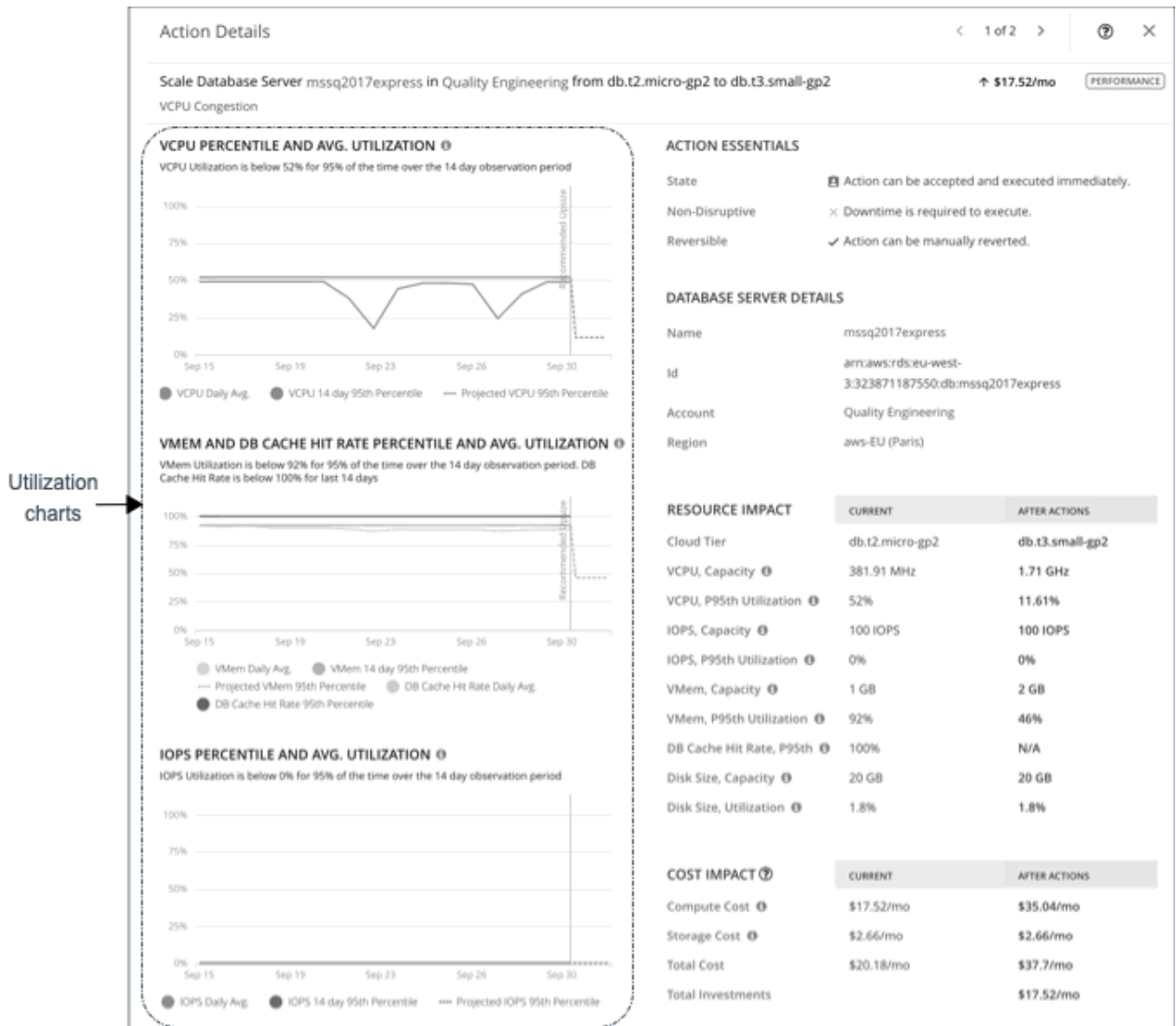
The table lists all the actions that are pending for Database Servers, and the savings or investments for each action.

| Scale Actions                              |         | 16             | Savings    | \$1,679.38/mo    | EXECUTE ACTIONS |                      | ↓             |                 |               |         |
|--------------------------------------------|---------|----------------|------------|------------------|-----------------|----------------------|---------------|-----------------|---------------|---------|
| Type to search                             |         |                |            |                  |                 |                      |               | ADD FILTER      |               |         |
| Database Server Name                       | Account | Non-Disrupt... | Reversible | Instance Type    | On-Demand       | New Instance Type    | New On-Demand | Action Category | Savings ↓     | Action  |
| <input type="checkbox"/> rds-mariamultiiaz |         | ×              | ✓          | db.t3.small-io1  | \$0.924/h       | db.t3.small-stand... | \$0.095/h     | SAVINGS         | ↓ \$604.80/rr | DETAILS |
| <input type="checkbox"/> testioautoscaling |         | ×              | ✓          | db.t3.micro-io1  | \$0.446/h       | db.t3.micro-gp2      | \$0.033/h     | SAVINGS         | ↓ \$301.00/rr | DETAILS |
| <input type="checkbox"/> rds-maria-io1     |         | ×              | ✓          | db.t3.micro-io1  | \$0.418/h       | db.t3.micro-stand... | \$0.031/h     | SAVINGS         | ↓ \$282.50/rr | DETAILS |
| <input type="checkbox"/> btc-dbs-1         |         | ×              | ✓          | db.m5.xlarge-io1 | \$0.514/h       | db.r6g.large-stan... | \$0.219/h     | SAVINGS         | ↓ \$215.20/rr | DETAILS |

For details on how Turbonomic calculates savings or investments, see [Estimated On-demand Costs for Cloud Database Servers](#).

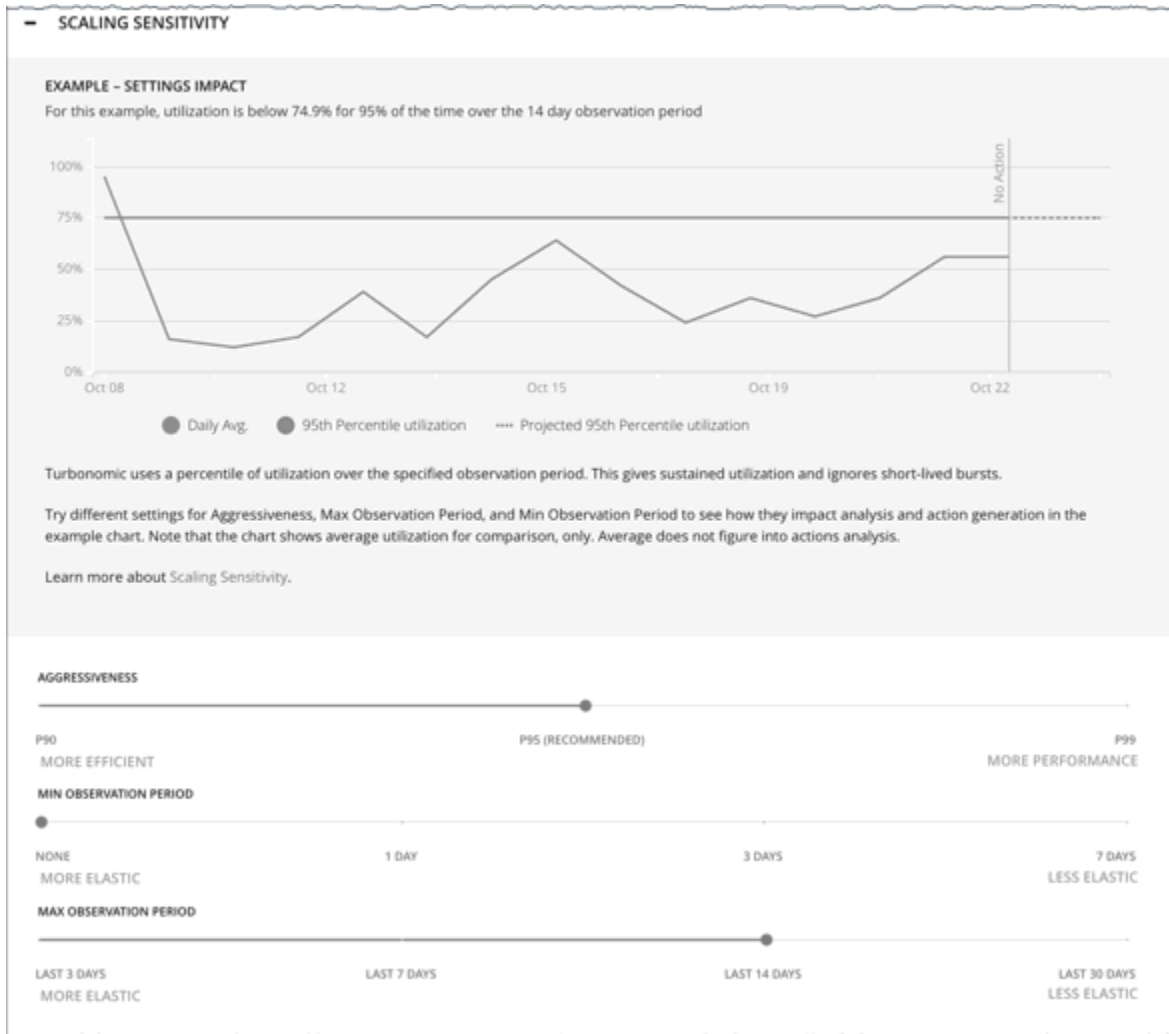
## Utilization Charts for Scale Actions

Turbonomic uses percentile calculations to measure resource utilization more accurately, and drive scaling actions that improve overall utilization and reduce costs. When you examine the details for a pending scale action on a Database Server, you will see charts that highlight *utilization percentiles* for a given observation period, and the projected percentiles after you execute the action.



The charts also plot *daily average utilization* for your reference. If you have previously executed scaling actions on the Database Server, you can see the resulting improvements in daily average utilization. Put together, these charts allow you to easily recognize utilization trends that drive Turbonomic's scaling recommendations.

You can set scaling constraints in Database Server policies to refine the percentile calculations.



For details, see [Scaling Sensitivity](#).

## Non-disruptive and Reversible Scaling Actions

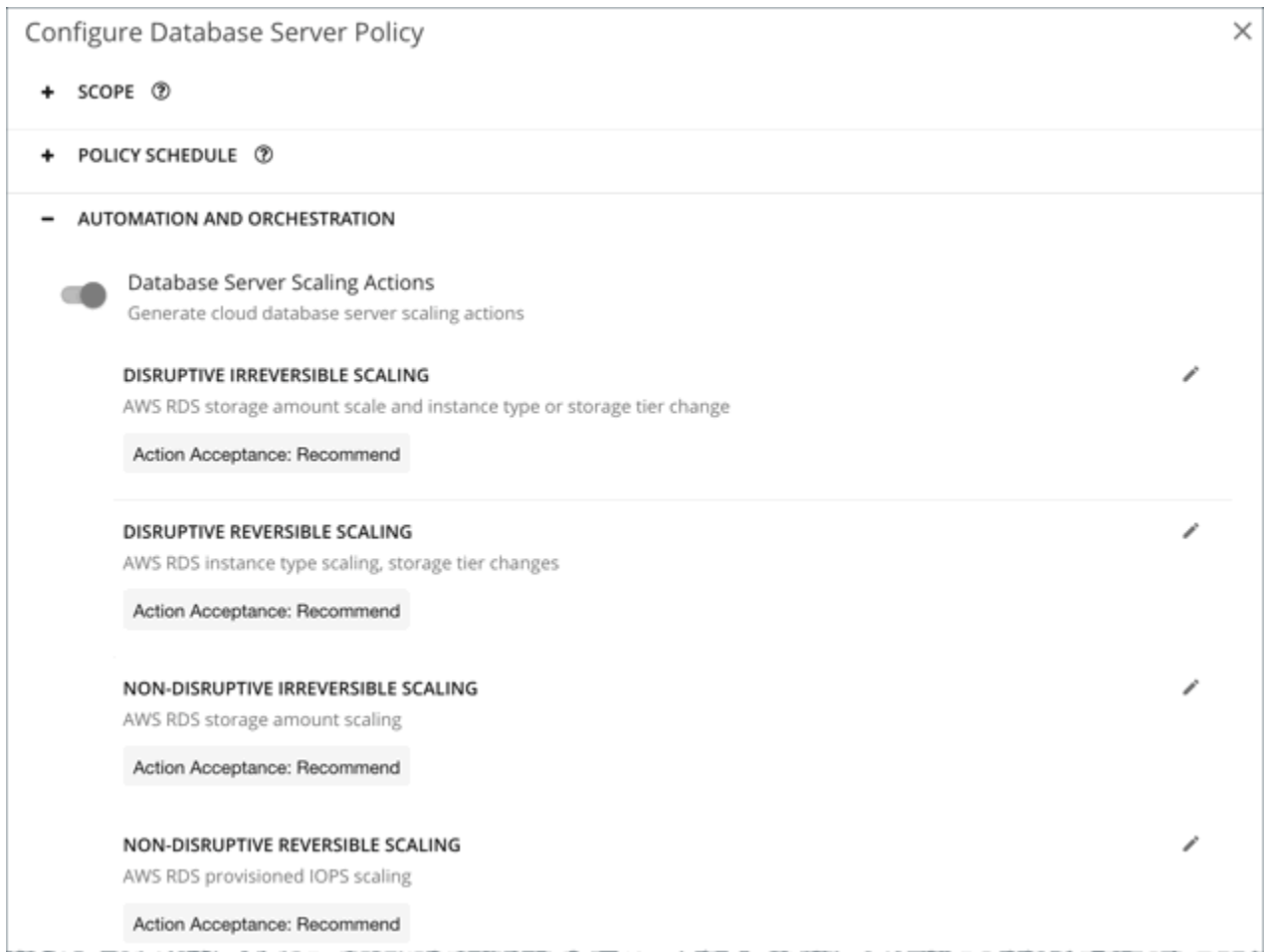
Turbonomic indicates whether a pending action is non-disruptive or reversible to help you decide how to handle the action.

| Scale Actions 16 Savings \$1,679.38/mo      |                          |         |                |            |                  |                |
|---------------------------------------------|--------------------------|---------|----------------|------------|------------------|----------------|
| <input type="text" value="Type to search"/> |                          |         |                |            |                  |                |
| <input type="checkbox"/>                    | Database Server Name     | Account | Non-Disruptive | Reversible | Instance Type    | On-Demand Cost |
| <input type="checkbox"/>                    | rds-mariamulti-az        |         | ✗              | ✓          | db.t3.small-io1  | \$0.924/h      |
| <input type="checkbox"/>                    | testioautoscalingenabled |         | ✗              | ✓          | db.t3.micro-io1  | \$0.446/h      |
| <input type="checkbox"/>                    | rds-maria-io1            |         | ✗              | ✓          | db.t3.micro-io1  | \$0.418/h      |
| <input type="checkbox"/>                    | btc-dbs-1                |         | ✗              | ✓          | db.m5.xlarge-io1 | \$0.514/h      |

The following table describes the disruptiveness and reversibility of the actions that Turbonomic recommends:

| Action                                                                                                         | Disruptive | Reversible |
|----------------------------------------------------------------------------------------------------------------|------------|------------|
| Scaling to a different instance type                                                                           | Yes        | Yes        |
| Scaling up storage amount                                                                                      | No         | No         |
| Scaling up storage access (provisioned IOPS)                                                                   | No         | Yes        |
| Scaling to a different storage type + Scaling up storage amount                                                | Yes        | No         |
| Scaling to a different storage type + Scaling up storage access (provisioned IOPS)                             | Yes        | Yes        |
| Scaling to a different storage type + Scaling up storage amount + Scaling up storage access (provisioned IOPS) | Yes        | No         |

You can set action acceptance modes in policies to specify the degree of automation for these actions.



## Unavailable Instance Types

A scale action could fail if the target instance type is unavailable in the availability zone for some reason. Your AWS environment might show the instance type as available, but when the scaling action executes, the following error displays in AWS:

```
Cannot modify the instance class because there are no instances of the requested class available in the current instance's availability zone. Please try your request again at a later time.
```

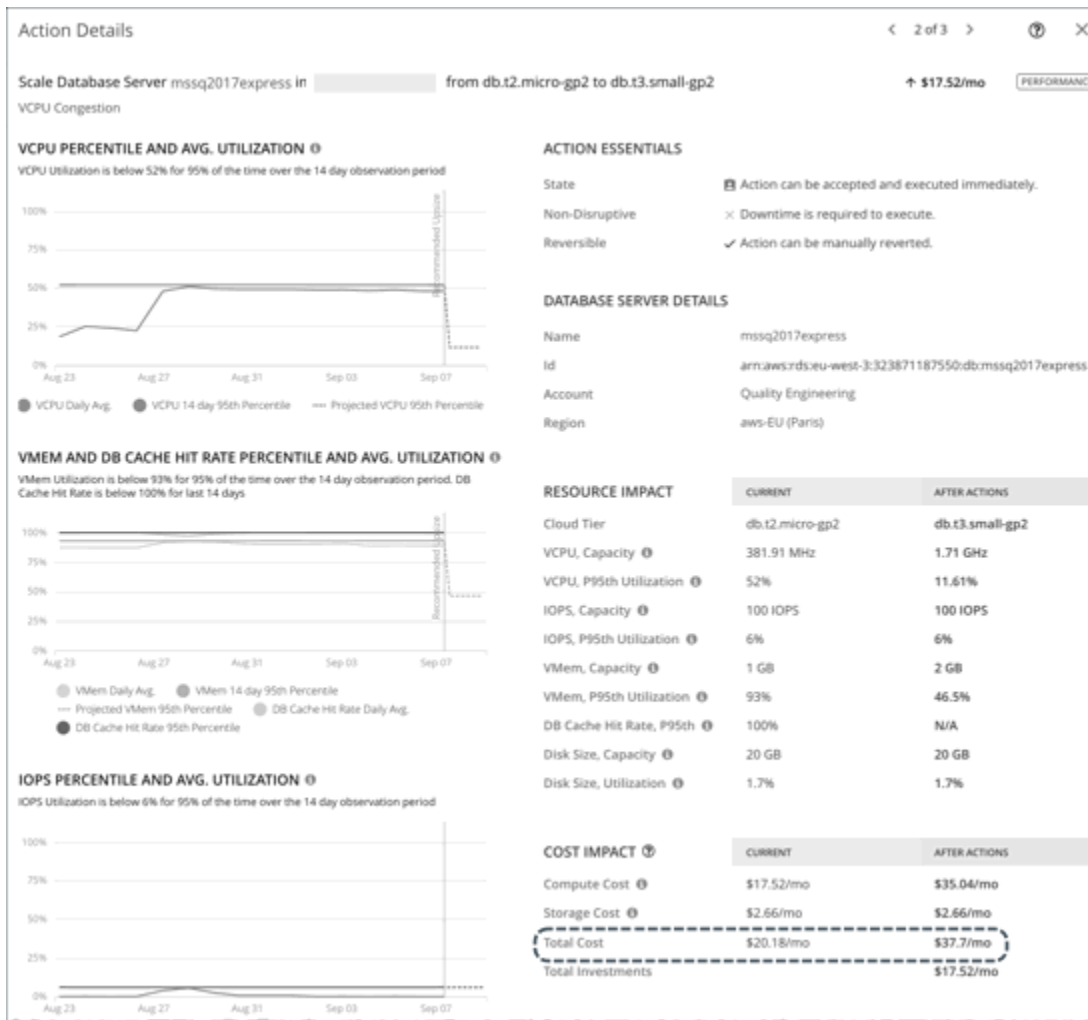
### Note:

For details about this error, see this [AWS page](#).

When this error occurs, Turbonomic modifies the default Database Server policy to exclude the instance type from its scaling list. When the Database Server is available again, Turbonomic adds it back to the scaling list. For details about this list, see [Cloud Instance Types](#).

## Estimated On-demand Costs for Cloud Database Servers

Turbonomic considers a variety of factors when calculating *Estimated On-demand Monthly Cost* for an AWS RDS Database Server.



## Non-Aurora Database Servers

### Cost Calculation

For non-Aurora Database Servers, the calculation for Estimated On-demand Monthly Cost can be expressed as follows:

$$(\text{On-demand Compute Rate} \times 730) + (\text{Provisioned Database Storage Rate} \times \text{Provisioned Database Storage Amount}) + (\text{Provisioned IOPS Rate} \times \text{Provisioned IOPS Amount}) = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Compute Rate** is the hourly cost for a Database Server's instance type  
You can obtain on-demand rates via [Amazon RDS Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.
- **Provisioned Database Storage Rate** is the hourly cost for a Database Server's provisioned database storage  
You can obtain provisioned database storage rates via [Amazon RDS Pricing](#).
- **Provisioned IOPS Rate** is the monthly cost for a Database Server's provisioned IOPS

Provisioned IOPS apply only to Database Servers on Provisioned IOPS SSD (io1) storage. You can obtain information about Provisioned IOPS SSD storage via the [RDS User Guide](#).

You can obtain provisioned IOPS rates via [Amazon RDS Pricing](#).

The listed items above impact cost calculations and the scaling decisions that Turbonomic makes. These decisions also rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

**Example**

Assume the following data for a pending scale action for SQL Server Express Edition (Single A-Z deployment):

|                                     | Current Values | Values After Action Execution |
|-------------------------------------|----------------|-------------------------------|
| On-demand Compute Rate              | \$0.024/hr     | \$0.048/hr                    |
| Provisioned Database Storage Rate   | \$0.133/hr     | \$0.133/hr                    |
| Provisioned Database Storage Amount | 20 GB          | 20 GB                         |
| Provisioned IOPS Rate               | \$0.00         | \$0.00                        |
| Provisioned IOPS Amount             | 0              | 0                             |

Turbonomic calculates the following:

- *Current* Estimated On-demand Monthly Cost:

$$(0.024 * 730) + (0.133 * 20) + (0.00 * 0) = 20.18$$

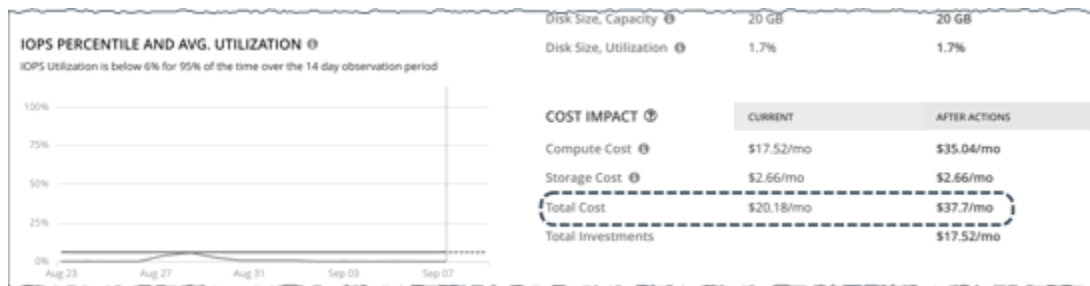
- Estimated On-demand Monthly Cost *after* executing the action:

$$(0.048 * 730) + (0.133 * 20) + (0.00 * 0) = 37.7$$

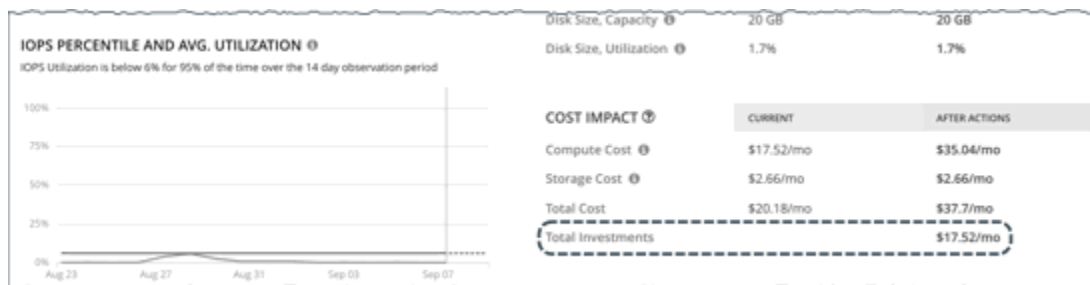
**Note:**

Turbonomic rounds the calculated values that it displays in the user interface.

The Estimated On-demand Monthly Cost is projected to increase from \$20.18/month to \$37.7/month, as shown in the Details section of the pending action.



Turbonomic treats the action as an investment and shows an estimated investment of \$17.52/month.



## Aurora Database Servers

### Cost Calculation

For Aurora Database Servers, the calculation for Estimated On-demand Monthly Cost can be expressed as follows:

$$(\text{On-demand Compute Rate} * 730) + (\text{Provisioned Database Storage Rate} * \text{Provisioned Database Storage Amount}) + (\text{I/O Request Rate} * (\text{Hourly Billed I/O Operation Count} * 730)) = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Compute Rate** is the hourly cost for a Database Server's instance type  
You can obtain on-demand rates via [Amazon Aurora Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.
- **Provisioned Database Storage Rate** is the hourly cost for a Database Server's provisioned database storage  
You can obtain provisioned database storage rates via [Amazon Aurora Pricing](#).
- **I/O Request Rate** is the cost per one million read/write I/O requests  
You can obtain I/O request rates via [Amazon Aurora Pricing](#).
- **Hourly Billed I/O Operation Count** is the average sum of read and write I/O operations per hour over the last month

The listed items above impact cost calculations. Except for I/O request rate, these items affect the actual scaling decisions that Turbonomic makes. These decisions also rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

### Example

Assume the following data for a pending scale action for Aurora MySQL-Compatible Edition:

|                                     | Current Values              | Values After Action Execution |
|-------------------------------------|-----------------------------|-------------------------------|
| On-demand Compute Rate              | \$0.164/hr                  | \$0.041/hr                    |
| Provisioned Database Storage Rate   | \$0.10/hr                   | \$0.10/hr                     |
| Provisioned Database Storage Amount | 100                         | 100                           |
| I/O Request Rate                    | \$0.20/one million requests | \$0.20/one million requests   |
| Hourly Billed I/O Operation Count   | 2000                        | 2000                          |

Turbonomic calculates the following:

- *Current* Estimated On-demand Monthly Cost:

$$(0.164 * 730) + (0.10 * 100) + ((0.20 / 1000000) * (2000 * 730)) = 130.01$$

- Estimated On-demand Monthly Cost *after* executing the action:

$$(0.041 * 730) + (0.10 * 100) + ((0.20 / 1000000) * (2000 * 730)) = 40.22$$

### Note:

Turbonomic rounds the calculated values that it displays in the user interface.



Since the Estimated On-demand Monthly Cost is projected to decrease from \$130.01/month to \$40.22/month, Turbonomic treats the action as a cost-saving measure and shows estimated savings of \$89.79/month.

## Cloud Database Server Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

For details about cloud Database Server actions, see [Cloud Database Server Actions](#) and [Non-disruptive and Reversible Scaling Actions](#).

| Action                              | Default Setting/Mode |
|-------------------------------------|----------------------|
| Database Server Scaling Actions     | On                   |
| Disruptive irreversible scaling     | Recommend            |
| Disruptive reversible scaling       | Recommend            |
| Non-disruptive irreversible scaling | Recommend            |
| Non-disruptive reversible scaling   | Recommend            |

### Scaling Sensitivity

Turbonomic uses a percentile of utilization over the specified observation period. This gives sustained utilization and ignores short-lived bursts.

Turbonomic uses these settings to calculate utilization percentiles for VCPU, VMEM and DB Cache Hit Rate, and IOPS. It then recommends actions to improve utilization based on the observed values for a given time period.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 95th percentile. The percentile utilization is the highest value that 95% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce capacity. Without using a percentile, Turbonomic never resizes below the recognized peak utilization. Assume utilization peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce resources for that Database Server.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single burst to 100%, but for 95% of the samples,

utilization never exceeded 50%. If you set **Aggressiveness** to 95th Percentile, then Turbonomic can see this as an opportunity to reduce resource allocation.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 99th Percentile – More performance. Recommended for critical Database Servers that need maximum guaranteed performance at all times, or those that need to tolerate sudden and previously unseen spikes in utilization, even though sustained utilization is low.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings. This assures performance while avoiding reactive peak sizing due to transient spikes, thus allowing you to take advantage of the elastic ability of the cloud.
- 90th Percentile – More efficiency. Recommended for Database Servers that can stand higher resource utilization.

By default, Turbonomic uses samples from the last 14 days. Use the **Max Observation Period** setting to adjust the number of days. To ensure that there are enough samples to analyze and drive scaling actions, set the **Min Observation Period**.

• **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 14 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the Database Server has fewer days' data then it uses all of the stored historical data.

You can make the following settings:

- Less Elastic – Last 30 Days
- Recommended – Last 14 Days
- More Elastic – Last 7 Days or Last 3 Days

Turbonomic recommends an observation period of 14 days so it can recommend scaling actions more often. Since Database Server scaling is minimally disruptive, scaling often should not introduce any noticeable performance risks.

• **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

- More Elastic – None
- Less Elastic – 1, 3, or 7 Days

## Cloud Instance Types

By default, Turbonomic considers all instance types currently available for scaling when making scaling decisions for Database Servers. However, you may have set up your Database Servers to *only scale to* or *avoid* certain instance types to reduce complexity and cost, or meet demand. Use this setting to identify the instance types that Database Servers can scale to.

### Note:

Turbonomic automatically discovers and enforces Database Server tier exclusions configured in your AWS environment. You do not need to configure these tier exclusions in policies. To see a list of tier exclusions that are currently enforced, set the scope to one or several Database Servers and click the **Policies** tab.

| Attribute            | Default Value |
|----------------------|---------------|
| Cloud Instance Types | None          |

Click **Edit** to set your preferences. In the new page that displays, expand a **cloud tier** (a family of instance types, such as *db.m1*) to see individual instance types and the resources allocated to them.

Select your preferred instance types or cloud tiers, or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

### Note:

This policy setting is not available in plans.

If you selected a cloud tier and the service provider deploys new instance types to that tier later, then those instance types will automatically be included in your policy. Be sure to review your policies periodically to see if new instance types have been added to a tier. If you do not want to scale to those instance types, update the affected policies.

## Scaling Target Utilization

This is the target utilization as a percentage of capacity.

| Attribute      | Default Value |
|----------------|---------------|
| VCPU           | 70            |
| VMEM           | 90            |
| IOPS           | 70            |
| Storage Amount | 90            |

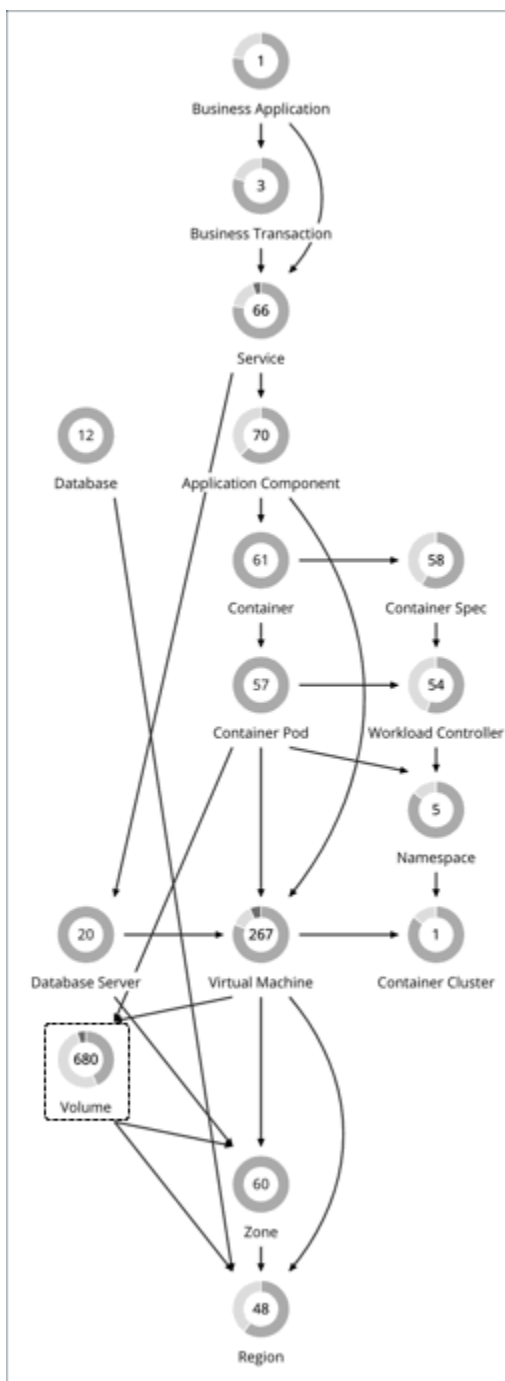
These advanced settings determine how much you would like a scope of workloads to utilize their resources. These are fixed settings that override the way Turbonomic calculates the optimal utilization of resources. You should only change these settings after consulting with Technical Support.

While these settings offer a way to modify how Turbonomic recommends actions, in most cases you should never need to use them. If you want to control how Turbonomic recommends actions to resize workloads, you can set the aggressiveness per the percentile of utilization, and set the length of the sample period for more or less elasticity on the cloud.

## Volume (Cloud)

A cloud volume is a storage device that you can attach to a VM. You can use an attached volume the same as a physical hard drive.

## Synopsis



## Synopsis

Budget:

A cloud volume gains its budget by selling resources to the VMs that it serves.

Provides:

Storage resources for VMs to use:

- Storage Access
- Storage Amount
- IO Throughput

|                     |                                               |
|---------------------|-----------------------------------------------|
| <b>Synopsis</b>     |                                               |
| Consumes:           | Storage services provided by Zones or Regions |
| Discovered through: | Cloud targets                                 |

## Monitored Resources

Turbonomic monitors the following resources for AWS/Azure volumes:

- Storage Amount

Storage Amount is the storage capacity (disk size) of a volume.

Turbonomic discovers Storage Amount, but does not monitor utilization.

For a Kubeturbo (container) deployment that includes volumes, Kubeturbo monitors Storage Amount utilization for the volumes. You can view utilization information in the Capacity and Usage chart.

- Storage Access (IOPS)

Storage Access, also known as IOPS, is the measurement of IOPS capacity that is in use.

- I/O Throughput

I/O Throughput is the measurement of I/O throughput capacity that is in use.

### Note:

Currently, Turbonomic does not monitor resources for GCP volumes. It only monitors their attachment state and then generates delete actions for unattached volumes.

## Actions

Turbonomic can recommend the following actions:

- **Scale**

Scale attached volumes to optimize performance and costs.

### Note:

This action is only supported for AWS and Azure volumes.

- **Delete**

Delete unattached volumes as a cost-saving measure. Turbonomic generates an action immediately after discovering an unattached volume.

## Scale Actions

Scale attached AWS/Azure volumes to optimize performance and costs.

### Note:

Currently, Turbonomic does not generate scale actions for GCP volumes.

Turbonomic can recommend:

- Scaling within the same tier (scale up or down), or from one tier to another

Examples:

- An action to scale down IOPS for a high performance volume, such as Azure Managed Ultra
- An action to scale a volume from the *io1* to the *gp2* tier

These actions can reduce costs significantly while continuing to assure performance. In addition, they are *non-disruptive* and *reversible*.

**Note:**

For details about action disruptiveness and reversibility, see [Non-disruptive and Reversible Scaling Actions](#).

- Scaling from one tier to another and then within the new tier, in a single action

For example, to achieve higher IOPS performance for VMs that are premium storage capable, you might see an action to scale the corresponding volume from *Standard* to *Premium*, and then scale up volume capacity from *32GB* to *256 GB*. This action increases costs and is *irreversible*, but is more cost effective than scaling up within the original tier.

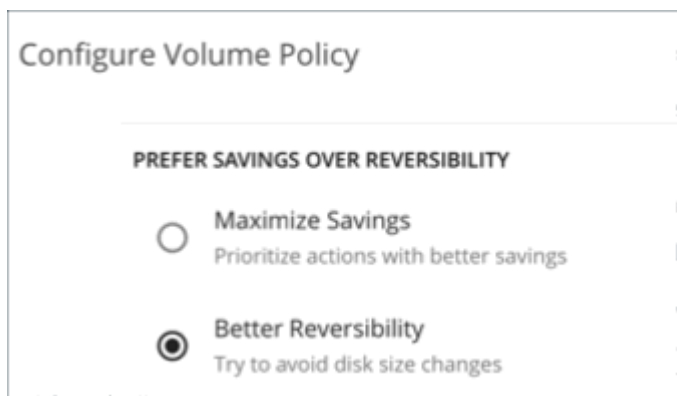
**Note:**

Not all VM types are premium storage capable. For details, see the [Azure Documentation](#).

When there are multiple disks attached to a volume, every volume scale action can potentially disrupt the same VM multiple times and some of the actions may fail due to concurrency. To mitigate these issues, Turbonomic identifies all volume actions associated with a particular VM and combines them into a single unit for execution, thus reducing disruptions and the chance of failures due to concurrency. This approach applies to scale actions in *Manual* or *Automated* mode.

You can create policies to control the scaling actions that Turbonomic generates.

- Turbonomic includes a policy setting that lets you choose between two outcomes – better savings (default) and better reversibility. If you choose reversibility, which can increase costs, Turbonomic will prioritize actions to change tiers whenever possible.



- Set scaling constraints if you want volumes to *only scale to* or *avoid* certain tiers. For details, see [Cloud Storage Tiers](#).

**Delete Actions**

Delete unattached volumes as a cost-saving measure. Turbonomic generates delete actions for AWS, Azure, and GCP volumes.

**Note:**

If you delete an Azure volume and then later deploy a new one with an identical name, charts will include historical data from the volume that you deleted.

**Exceptions for Azure Site Recovery and Azure Backup Volumes**

Turbonomic discovers Azure Site Recovery and Azure Backup volumes when you add Azure targets. Even though these volumes are always unattached, Turbonomic will never recommend deleting them because they are critical to business continuity and disaster recovery.

To identify Azure Site Recovery volumes, Turbonomic checks an Azure resource called [Recovery Services vault](#), which includes information specific to the volumes. It also checks for the `ASR-ReplicaDisk` tag, which Azure automatically assigns to the volumes.

For Azure Backup volumes, Turbonomic checks for the `RSVaultBackup` tag.

It is important that you do not remove these tags. If these tags have been removed for some reason, create a volume policy for the affected volumes and disable the *Delete* action in that policy.

## Action Execution for Locked Volumes

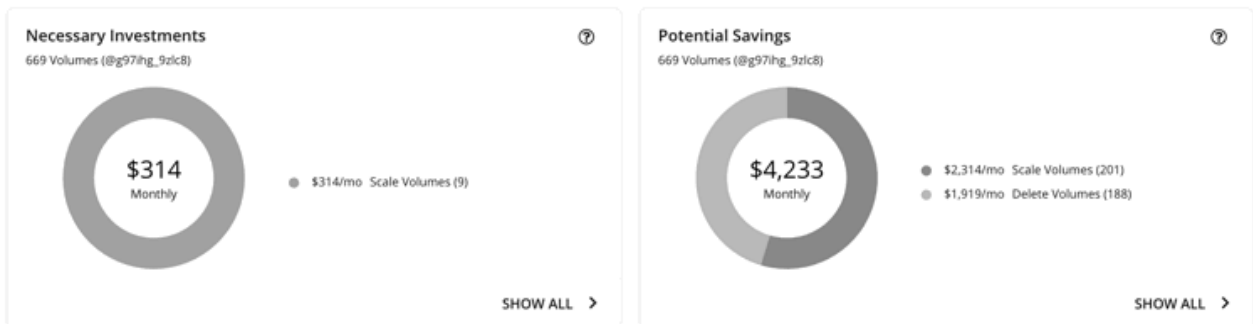
For Azure environments, Turbonomic can recommend scale and delete actions for locked volumes, but the lock level configured for the volumes may prevent some actions from executing.

- For volumes with the ReadOnlY lock, both scale and delete actions are *not* executable.
- For volumes with the CanNotDeLeTe lock, delete actions are *not* executable, but scale actions are executable.

You must sign in to Azure and then remove the locks for the affected volumes before you can execute actions. To identify the specific locks that you need to remove, open the Action Details page for a pending volume action and see the **Execution Prerequisites** section.

## Volume Actions in Charts

Use the Necessary Investments and Potential Savings charts to view pending volume actions. These charts show total monthly investments and savings, assuming you execute all the actions.



Click **Show All** for each chart to review and execute the actions.

The table shows the following:

- Actions that are pending for each volume
- Savings or investments for each volume
- For *Delete* actions in the Potential Savings chart:

| Potential Savings                                                   |                 |                      |                 |        |                 |            |                 |         |             |         |
|---------------------------------------------------------------------|-----------------|----------------------|-----------------|--------|-----------------|------------|-----------------|---------|-------------|---------|
| DELETE ^                                                            |                 |                      |                 |        |                 |            |                 |         |             |         |
| AWS AZURE                                                           |                 |                      |                 |        |                 |            |                 |         |             |         |
| Volumes (191) Delete Actions 189 Savings \$2,033/mo EXECUTE ACTIONS |                 |                      |                 |        |                 |            |                 |         |             |         |
| SCALE ^                                                             |                 |                      |                 |        |                 |            |                 |         |             |         |
| Volumes (157)                                                       |                 |                      |                 |        |                 |            |                 |         |             |         |
| Volume ID                                                           | Subscription    | Tier Type            | Size            | State  | Days Unattached | Image Disk | Action Category | Savings | Action      |         |
| <input type="checkbox"/>                                            | aks-agentpool-3 | Pay-As-You-Go - Prod | Managed Premium | 30 GiB | Unattached      | 4          | /Subscripti...  | SAVINGS | ↓ \$5.28/mo | DETAILS |
| <input type="checkbox"/>                                            | aks-agentpool-4 | Pay-As-You-Go - Prod | Managed Premium | 30 GiB | Unattached      | 4          | /Subscripti...  | SAVINGS | ↓ \$5.28/mo | DETAILS |
| <input type="checkbox"/>                                            | aks-agentpool-8 | Pay-As-You-Go - Prod | Managed Premium | 30 GiB | Unattached      | 4          | /Subscripti...  | SAVINGS | ↓ \$5.28/mo | DETAILS |
| <input type="checkbox"/>                                            | aks-agentpool-7 | Pay-As-You-Go - Prod | Managed Premium | 30 GiB | Unattached      | 4          | /Subscripti...  | SAVINGS | ↓ \$5.28/mo | DETAILS |

– Number of days a volume has been unattached

This information helps you decide whether to take the action.

Turbonomic polls your cloud volumes every 6 hours, and then records their state (attached or unattached) at the time of polling. It does not account for changes in state between polls.

For newly unattached volumes, Turbonomic shows a dash symbol (-) if a volume has been unattached within the last 6 hours. A value of 0 (zero) means that a volume has been unattached within the last 24 hours.

Once Turbonomic discovers an unattached volume, it immediately recommends that you delete it. If a currently unattached volume is not deleted and is subsequently discovered as attached, Turbonomic removes the *Delete* action attached to it, and then resets the unattached period.

**Note:**

For volumes that have been deleted from the cloud provider and are no longer discoverable, Turbonomic removes them from its records after 15 days.

To see the last known VM attached to the volume, click **DETAILS**.

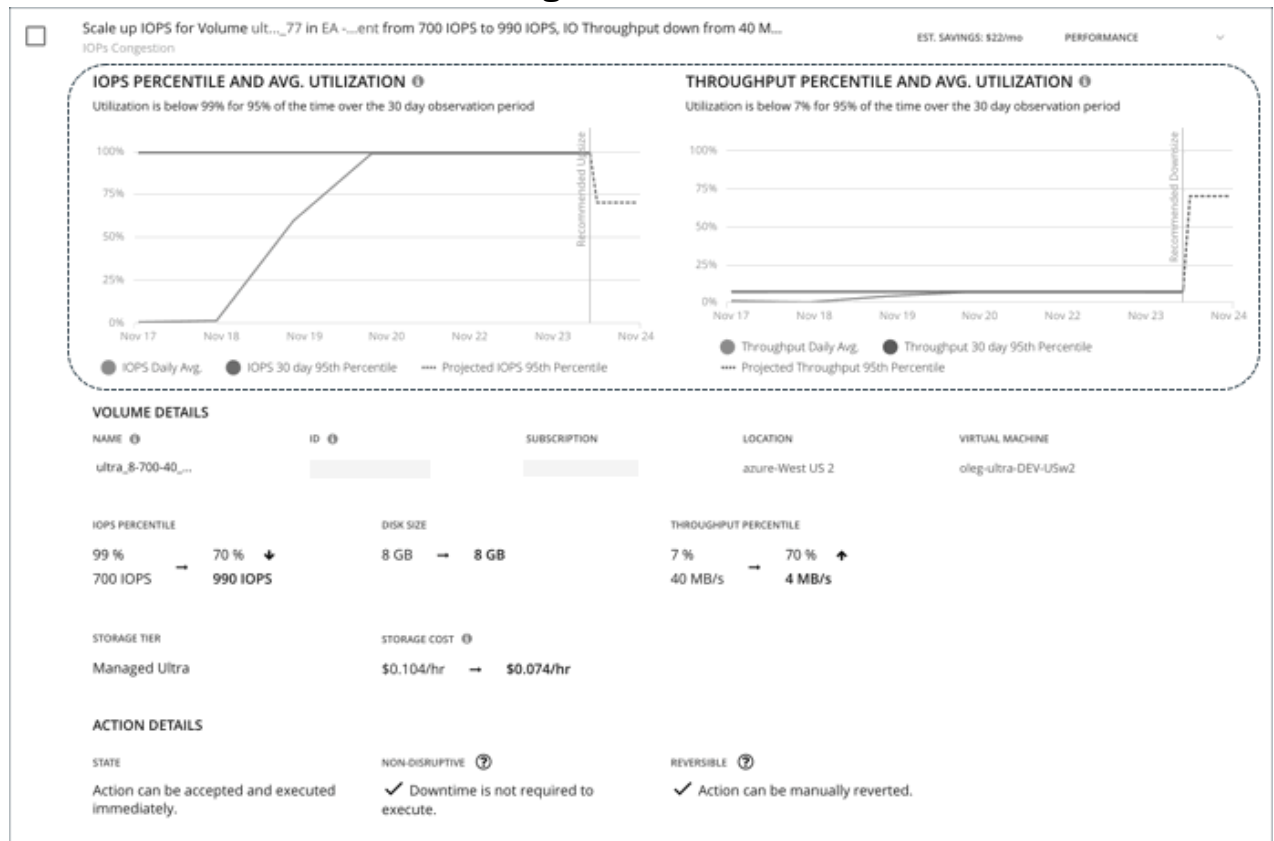
- For *Scale* actions in the Potential Savings or Necessary Investments chart:

| Potential Savings                    |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
|--------------------------------------|---------|----------------|------------|-------------|-----------|------|----------|-------------|-----------|----------|----------|-----------------|------------|---------|
| SCALE ^                              |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| Scale Actions 201 Savings \$2,314/mo |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| EXECUTE ACTIONS                      |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| Volumes (201)                        |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| DELETE ^                             |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| Volumes (188)                        |         |                |            |             |           |      |          |             |           |          |          |                 |            |         |
| Volume Name                          | Account | Non-Disruptive | Reversible | Tier        | Disk Size | IOPS | Cost     | New Tier    | Disk Size | New IOPS | New Cost | Action Category | Savings    | Action  |
| snap_irmel_dotn                      | Dev     | ✗              | ✓          | Managed ... | 1 TB      | 5000 | \$149/mo | Managed ... | 1 TB      | 500      | \$41/mo  | SAVINGS         | ↓ \$108/mo | DETAILS |
| PTEricDisks2_Di                      | Prod    | ✓              | ✓          | Managed ... | 256 GB    | 1500 | \$284/mo | Managed ... | 256 GB    | 2143     | \$180/mo | PERFORMANCE     | ↓ \$104/mo | DETAILS |
| SQLServerDyna'                       | Dev     | ✗              | ✓          | Managed ... | 1 TB      | 5000 | \$135/mo | Managed ... | 1 TB      | 500      | \$41/mo  | SAVINGS         | ↓ \$94/mo  | DETAILS |
| SQLServerDyna'                       | Dev     | ✗              | ✓          | Managed ... | 1 TB      | 5000 | \$135/mo | Managed ... | 1 TB      | 500      | \$41/mo  | SAVINGS         | ↓ \$94/mo  | DETAILS |
| SQLServerTestV                       | Dev     | ✗              | ✓          | Managed ... | 1 TB      | 5000 | \$135/mo | Managed ... | 1 TB      | 500      | \$41/mo  | SAVINGS         | ↓ \$94/mo  | DETAILS |
| SQLServerTestV                       | Dev     | ✗              | ✓          | Managed ... | 1 TB      | 5000 | \$135/mo | Managed ... | 1 TB      | 500      | \$41/mo  | SAVINGS         | ↓ \$94/mo  | DETAILS |

- Whether actions are non-disruptive or reversible
- Changes the actions will effect (for example, changes in tiers and/or resource allocations)

When you click the **DETAILS** button for a scaling action, you will see utilization charts that help explain the reason for the action.

**Utilization Charts for Volume Scaling Actions**





Turbonomic uses percentile calculations to measure IOPS and throughput more accurately, and drive scaling actions that improve overall utilization and reduce costs. When you examine the details for a pending scaling action on a volume, you will see charts that highlight resource *utilization percentiles* for a given observation period, and the projected percentiles after you execute the action.

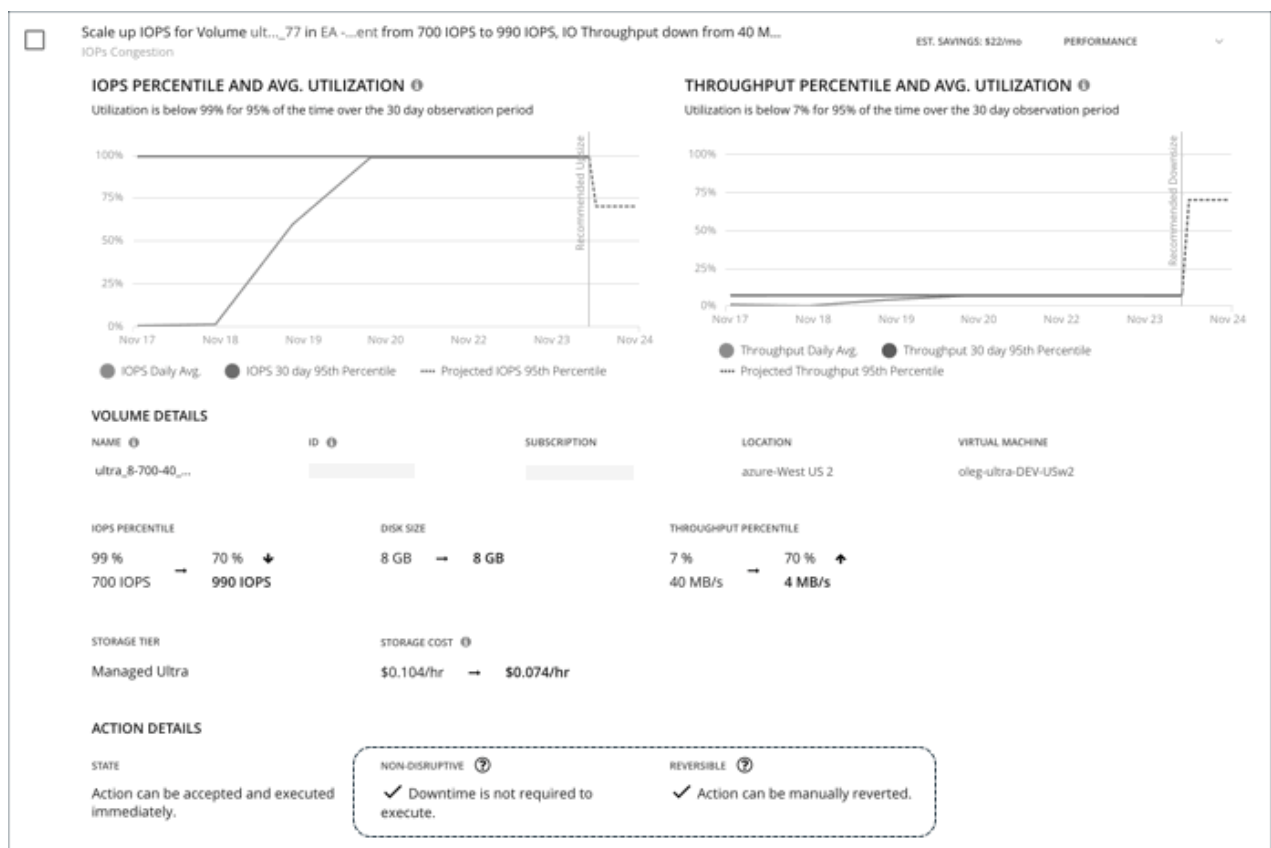
The charts also plot *daily average utilization* for your reference. If you have previously executed scaling actions on the volume, you can see the resulting improvements in daily average utilization. Put together, these charts allow you to easily recognize utilization trends that drive Turbonomic's scaling recommendations.

**Note:**

You can set scaling sensitivity in volume policies to refine the percentile calculations. For details, see [Scaling Sensitivity](#).

**Non-disruptive and Reversible Scaling Actions**

Turbonomic indicates whether a pending action is non-disruptive or reversible.



• **Non-disruptive**

Executing storage scaling actions can sometimes be disruptive if the VM must be rebooted to execute a storage change. For example, Azure Standard and Premium scaling actions are *disruptive*. When a storage action is disruptive, expect a single reboot (usually 2-3 minutes of downtime).

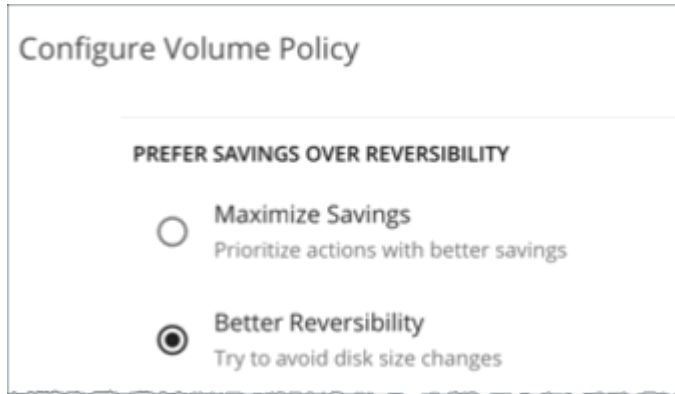
The following scaling actions are *non-disruptive*:

- Scaling IOPS and throughput on Azure Ultra storage
- All scaling actions on AWS storage

• **Reversible**

Executing storage scaling actions can sometimes be irreversible if the volume must grow in size to subsequently increase IOPS or throughput capacity. In this case, shrinking that volume's size later

would not be possible. If you prefer reversible volume actions, create a volume policy and choose **Better Reversibility**.



## Cloud Volume Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

For details about cloud volume actions, see [Cloud Volume Actions](#).

| Action | Default Mode |
|--------|--------------|
| Scale  | Manual       |
| Delete | Manual       |

#### Prefer Savings Over Reversibility

Executing storage scaling actions can sometimes be irreversible if the volume must grow in size to subsequently increase IOPS or throughput capacity. In this case, shrinking that volume's size later would not be possible. If you prefer reversible volume actions, create a volume policy and choose **Better Reversibility**.

#### Scaling Sensitivity

Turbonomic uses a percentile of utilization over the specified observation period. This gives sustained utilization and ignores short-lived bursts.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

Turbonomic uses Aggressiveness when evaluating IOPS and throughput.

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 95th percentile. The percentile utilization is the highest value that 95% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce capacity. Without using a percentile, Turbonomic never scales below the recognized peak utilization. Assume utilization peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce resources for that volume.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single burst to 100%, but for 95% of the samples, utilization never exceeded 50%. If you set **Aggressiveness** to 95th Percentile, then Turbonomic can see this as an opportunity to reduce resource allocation.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 99th or 100th Percentile – More performance. Recommended for critical volumes that need maximum guaranteed performance at all times, or those that need to tolerate sudden and previously unseen spikes in utilization, even though sustained utilization is low.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings. This assures performance while avoiding reactive peak sizing due to transient spikes, thus allowing you to take advantage of the elastic ability of the cloud.
- 90th Percentile – More efficiency. Recommended for volumes that can stand higher resource utilization.

By default, Turbonomic uses samples from the last 30 days. Use the **Max Observation Period** setting to adjust the number of days.

• **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 30 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the volume has fewer days' data then it uses all of the stored historical data.

Choose from the following settings:

- Less Elastic – Last 90 Days
- Recommended – Last 30 Days
- More Elastic – Last 7 Days

• **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

Choose from the following settings:

- More Elastic – None
- Less Elastic – 1, 3, or 7 Days

## Scaling Target IOPS/Throughput Utilization

This is the target utilization as a percentage of capacity.

| Attribute                                  | Default Value |
|--------------------------------------------|---------------|
| Scaling Target IOPS/Throughput Utilization | 70            |

## Cloud Storage Tiers

By default, Turbonomic considers all storage tiers currently available for scaling when making scaling decisions for volumes. However, you may have set up your cloud volumes to *only scale to* or *avoid* certain tiers to reduce complexity and cost, or meet demand. Use this setting to identify the tiers that volumes can scale to.

| Attribute           | Default Value |
|---------------------|---------------|
| Cloud Storage Tiers | None          |

Click **Edit** to set your preferences. In the new page that displays, select your preferred cloud tiers or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

## Database (Cloud)

Turbonomic discovers SQL Databases and dedicated SQL pools from your Azure targets, and represents them as Database entities in the supply chain.

### Note:

AWS RDS databases appear as *Database Server* entities in the supply chain. For details, see [Database Server \(Cloud\)](#).

- [SQL Databases](#)

Turbonomic discovers the resources on *individual databases* that are managed under both the DTU (Database Transaction Unit) and vCore pricing models.

- DTU Pricing Model

In the DTU model, Azure bundles CPU, memory, and IOPS as a single DTU metric. Turbonomic actions on these databases consider both DTU and storage utilization.

- vCore Pricing Model

In the vCore model, analysis can track CPU, memory, IOPS, and throughput metrics in isolation. Turbonomic actions on these databases are driven by CPU, memory, IOPS, throughput and storage utilization.

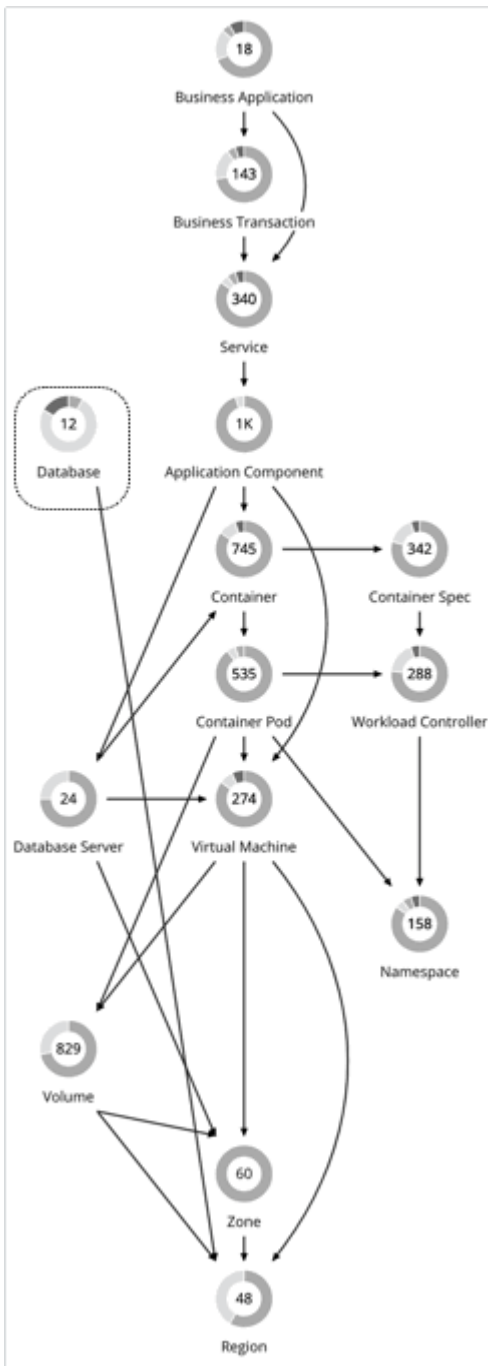
### Note:

For more information about the DTU and vCore models, see the .

- [Dedicated SQL Pools](#) (for Azure Synapse Analytics)

A dedicated SQL pool represents a collection of analytic resources that Azure provisions when you use Synapse SQL. Compute capacity for a dedicated SQL pool is expressed in Data Warehousing Units (DWU). DWU represents CPU, memory, and IO Throughput bundled as a single commodity.

## Synopsis



### Synopsis

Provides:

Transactions to end users and analytic resources to Azure Synapse Analytics

## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Consumes:           | <ul style="list-style-type: none"><li>• DTU Pricing Model:<br/>DTU and storage resources in an Azure region</li><li>• vCore Pricing Model:<br/>vCPU, vMem, IOPS, throughput, and storage resources in an Azure region</li><li>• Dedicated SQL Pool:<br/>DWU and storage resources in an Azure region</li></ul> |
| Discovered through: | Azure targets                                                                                                                                                                                                                                                                                                  |

## Database Groups

We highly recommend creating groups in Turbonomic so you can manage SQL databases and dedicated SQL pools with ease. When you set the scope to a particular group, you can evaluate pending actions, view metrics, and assign policies as usual. You can also view the status of individual members. For example, you can see if a dedicated SQL pool is currently active or suspended.

To create groups, go to **Settings > Groups**, and then click **New Group**. In the page that displays, choose **Dynamic** (recommended) and then set the Pricing Model filter as follows:

- **DTU** and/or **vCore** – use to create a group of SQL databases
- **DWU** – use to create a group of dedicated SQL pools

## Monitored Resources

The resources that Turbonomic can monitor depend on the pricing model in place for the given database entity.

- DTU Pricing Model
  - DTU  
DTU is the measurement of compute capacity for the database. DTU represents CPU, memory, and IOPS/IO Throughput bundled as a single commodity.
  - Storage Amount  
Storage Amount is the measurement of storage capacity utilized by the entity.
- vCore Pricing Model
  - Virtual Memory (VMem)  
Virtual Memory is the measurement of memory utilized by the entity.
  - Virtual CPU (VCPU)  
Virtual CPU is the measurement of CPU utilized by the entity.
  - Storage Amount  
Storage Amount is the measurement of storage capacity utilized by the entity.
  - Storage Access (IOPS)  
Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.
  - I/O Throughput  
I/O Throughput is the measurement of an entity's throughput to the underlying storage.
- Dedicated SQL Pool

- DWU

DWU (Data Warehousing Unit) is the measurement of compute capacity for the dedicated SQL pool. DWU represents CPU, memory, and IO Throughput bundled as a single commodity.

- Storage Amount

Storage Amount is the measurement of storage capacity utilized by the entity.

- Connection

Connection is the measurement of database connections utilized by applications.

## Actions

### Scale (SQL Database)

- DTU Pricing Model

Scale DTU and storage resources to optimize performance and costs.

- vCore Pricing Model

Scale vCPU, vMem, IOPS, throughput and storage resources to optimize performance and costs.

### Suspend (Dedicated SQL Pool)

Suspend an idle dedicated SQL pool (used for Azure Synapse Analytics) to reduce compute costs.

#### Note:

Currently, Turbonomic does not generate actions to start a suspended pool. You can manually start a pool from Azure.

## Scale Actions for SQL Databases

Turbonomic database scaling actions aim to increase resource utilization and reduce costs while complying with business policies.

Turbonomic drives scaling actions based on the utilization of resources, and treats the following limits as constraints when it makes scaling decisions:

- Maximum concurrent sessions

This is the maximum number of database connections at a time.

- Maximum concurrent workers

This is the maximum number of database processes that can handle queries at a time.

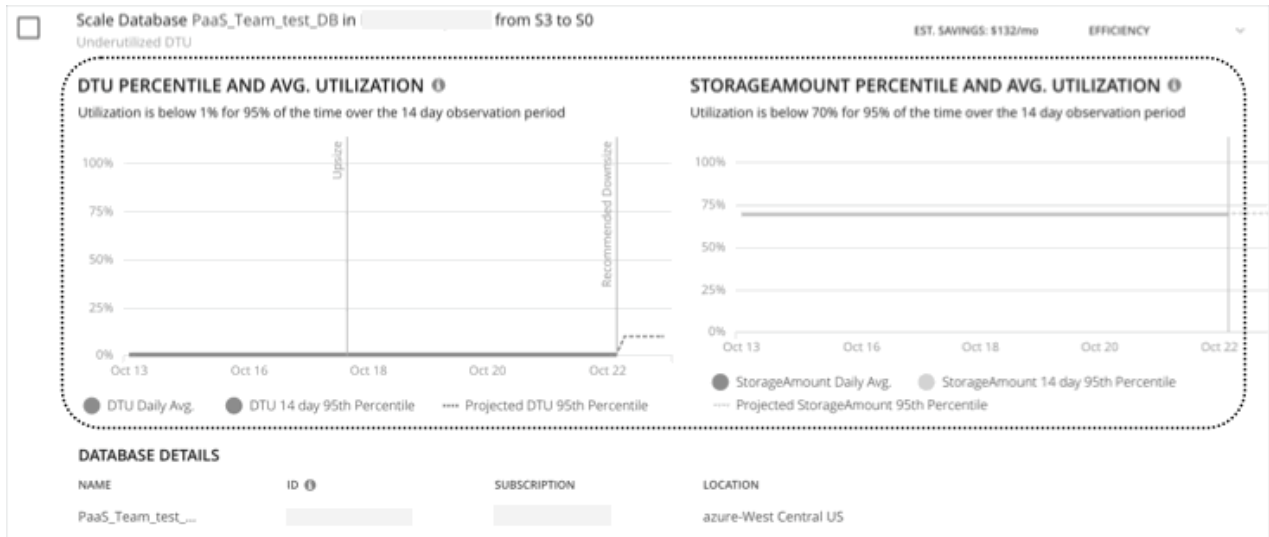
Points to consider:

- Turbonomic will *not* recommend:
  - Scaling from one pricing model to another
  - Scaling vCore databases to instance types running Gen4 hardware. This hardware generation has been retired and pricing information can no longer be retrieved via the Azure API.
  - Scaling vCore databases on the [serverless compute tier](#)
  - Scaling provisioned memory for vCore databases on the Hyperscale service tier. VMem utilization data is currently unavailable for Hyperscale due to an issue in the Azure API.
- On DTU databases, a single action can scale both DTU and storage. On vCore databases, a single action can scale vCPU, vMem, IOPS, throughput, and storage.
- In some cases, Turbonomic might recommend scaling up storage, even if there is no storage pressure on the database, to take advantage of storage provided at no extra cost. For example, Turbonomic might recommend scaling from the *S3* to the *S0* tier because of low DTU and storage utilization. Since the *S0* tier includes 250 GB of storage at no extra cost, Turbonomic will also recommend scaling up to this

storage amount. If you want to scale DTU but keep the storage amount unchanged, adjust the values for aggressiveness (percentile) and observation period in your database policies.

## Utilization Charts for Scale Actions

Turbonomic uses percentile calculations to measure resource utilization, and drive scaling actions that improve overall utilization and reduce costs. When you examine the details for a pending scaling action on a database, you will see charts that highlight resource *utilization percentiles* for a given observation period, and the projected percentiles after you execute the action.



The charts also plot *daily average utilization* for your reference. If you have previously executed scaling actions on the database, you can see the resulting improvements in daily average utilization. Put together, these charts allow you to easily recognize utilization trends that drive Turbonomic's scaling recommendations.

### Note:

You can set scaling constraints in database policies to refine the percentile calculations. For details, see [Aggressiveness and Observation Period](#).

## Non-disruptive and Reversible Scaling Actions

All scaling actions shown in the Action Center view and Action Details page are non-disruptive and reversible.

For actions to scale vCore databases from General Purpose or Business Critical to Hyperscale, there are certain caveats associated with reversing such actions. To learn more, see the [Azure documentation](#).

## Suspend Actions for Dedicated SQL Pools

When Turbonomic discovers an idle dedicated SQL pool (i.e., a pool with no active connections for a specific period of time), it will immediately recommend that you suspend the pool as a cost-saving measure. This action stops the compute instance for the SQL pool, but keeps storage running. In the Action Details page for a specific action, the compute cost is 0 (zero) *after* suspension, while storage cost remains the same.



| Action Details                                                                 |                                                  |                         |               |
|--------------------------------------------------------------------------------|--------------------------------------------------|-------------------------|---------------|
| Suspend Database paassqpool in [redacted]<br>Improve infrastructure efficiency |                                                  |                         |               |
| <b>ACTION ESSENTIALS</b>                                                       |                                                  | <b>DATABASE DETAILS</b> |               |
| State                                                                          | Action can be accepted and executed immediately. |                         |               |
| Non-Disruptive                                                                 | Downtime is not required to execute.             |                         |               |
| Reversible                                                                     | Action can be manually reverted.                 |                         |               |
| <b>RESOURCE IMPACT</b>                                                         | <b>CURRENT</b>                                   | <b>AFTER ACTIONS</b>    |               |
| DWU, Capacity                                                                  | 100                                              | -                       |               |
| DWU, Utilization                                                               | 0%                                               | -                       |               |
| Storage, Capacity                                                              | 1.34 GB                                          | 1.34 GB                 | -             |
| Storage, Utilization                                                           | 100%                                             | 100%                    | -             |
| <b>COST IMPACT</b>                                                             | <b>CURRENT</b>                                   | <b>AFTER ACTIONS</b>    |               |
| Compute Cost                                                                   | \$876.00/mo                                      | \$0.00/mo               | ↓ \$876.00/mo |
| Storage Cost                                                                   | \$23.00/mo                                       | \$23.00/mo              | -             |
| Total Cost                                                                     | \$899.00/mo                                      | \$23.00/mo              | ↓ \$876.00/mo |
| Total Savings                                                                  |                                                  | \$876.00/mo             |               |
| Name                                                                           | paassqpool                                       |                         |               |
| Id                                                                             | paassqpool                                       |                         |               |
| Subscription                                                                   | [redacted]                                       |                         |               |
| Workspace                                                                      | paas-synapse-workspace                           |                         |               |
| Pool Type                                                                      | Dedicated SQL pool                               |                         |               |
| Pricing Model                                                                  | DWU                                              |                         |               |
| Idle Time                                                                      | 14 days 12 hours 47 minutes                      |                         |               |

**Note:**

To calculate the current compute cost, Turbonomic retrieves the hourly on-demand cost from Azure and then multiplies the result by 730, which represents the number of hours per month that the product uses to estimate monthly costs.

If a currently idle pool is not suspended and is subsequently discovered as active, Turbonomic removes the suspend action attached to it.

Suspend actions include the 'Idle Time' information that indicates how long a dedicated SQL pool has been idle. By default, Turbonomic will generate suspend actions if dedicated SQL pools have been idle for at least one hour.

| Action Details                                                                 |                                                  |                         |               |
|--------------------------------------------------------------------------------|--------------------------------------------------|-------------------------|---------------|
| Suspend Database paassqpool in [redacted]<br>Improve infrastructure efficiency |                                                  |                         |               |
| <b>ACTION ESSENTIALS</b>                                                       |                                                  | <b>DATABASE DETAILS</b> |               |
| State                                                                          | Action can be accepted and executed immediately. |                         |               |
| Non-Disruptive                                                                 | Downtime is not required to execute.             |                         |               |
| Reversible                                                                     | Action can be manually reverted.                 |                         |               |
| <b>RESOURCE IMPACT</b>                                                         | <b>CURRENT</b>                                   | <b>AFTER ACTIONS</b>    |               |
| DWU, Capacity                                                                  | 100                                              | -                       |               |
| DWU, Utilization                                                               | 0%                                               | -                       |               |
| Storage, Capacity                                                              | 1.34 GB                                          | 1.34 GB                 | -             |
| Storage, Utilization                                                           | 100%                                             | 100%                    | -             |
| <b>COST IMPACT</b>                                                             | <b>CURRENT</b>                                   | <b>AFTER ACTIONS</b>    |               |
| Compute Cost                                                                   | \$876.00/mo                                      | \$0.00/mo               | ↓ \$876.00/mo |
| Storage Cost                                                                   | \$23.00/mo                                       | \$23.00/mo              | -             |
| Total Cost                                                                     | \$899.00/mo                                      | \$23.00/mo              | ↓ \$876.00/mo |
| Total Savings                                                                  |                                                  | \$876.00/mo             |               |
| Name                                                                           | paassqpool                                       |                         |               |
| Id                                                                             | paassqpool                                       |                         |               |
| Subscription                                                                   | [redacted]                                       |                         |               |
| Workspace                                                                      | paas-synapse-workspace                           |                         |               |
| Pool Type                                                                      | Dedicated SQL pool                               |                         |               |
| Pricing Model                                                                  | DWU                                              |                         |               |
| Idle Time                                                                      | 14 days 12 hours 47 minutes                      |                         |               |

You can control the suspend actions that Turbonomic generates, based on your preferred idle time value.

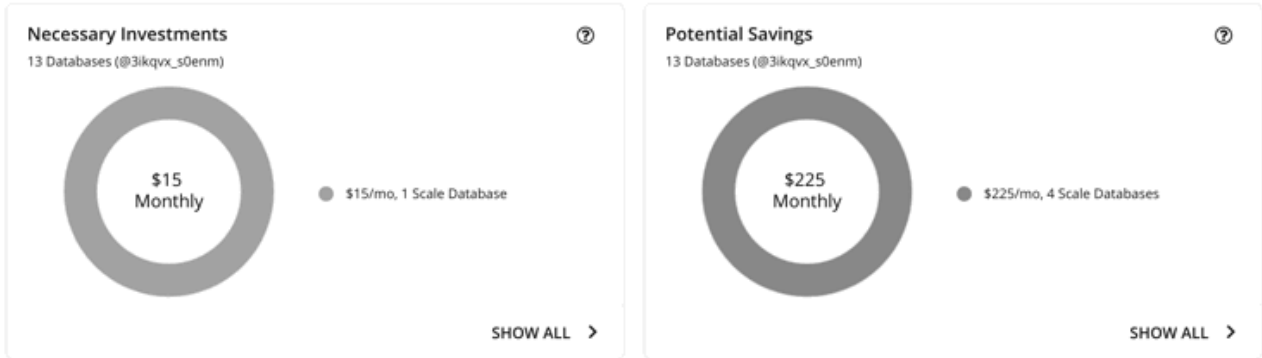
For example, if you want Turbonomic to only generate suspend actions for dedicated SQL pools that have been idle for at least one day, perform these steps:

1. Create a dynamic group of Databases and set the 'Hours Idle' filter to `Hours Idle <= 24`.

2. Create a custom Database policy, set the scope to the group that you just created, and then *disable* suspend actions in that policy.

## Actions in Charts

Use the Necessary Investments and Potential Savings charts to view pending database actions. These charts show total monthly investments and savings, assuming you execute all the actions.



Click **Show All** for each chart to review and execute the actions.

The table shows the following:

- Actions that are pending for each database
- Savings or investments for each database

## Estimated On-demand Costs for Cloud Databases

Turbonomic considers a variety of factors when calculating *Estimated On-demand Monthly Cost* for an Azure SQL Database.

| Database Server                   |                                                                                                           |                    |
|-----------------------------------|-----------------------------------------------------------------------------------------------------------|--------------------|
| Database Server                   | [Redacted]                                                                                                |                    |
| Pricing Model                     | DTU                                                                                                       |                    |
| Tags                              | turbo_owner: [Redacted]<br>turbo_comment: This is part of the test environment f...<br>turbo_lifetime: 10 |                    |
| RESOURCE IMPACT                   |                                                                                                           |                    |
|                                   | CURRENT                                                                                                   | AFTER ACTIONS      |
| Cloud Tier                        | S3                                                                                                        | S0                 |
| DTU, Capacity                     | 100                                                                                                       | 10                 |
| DTU, P95th Utilization            | 1%                                                                                                        | 10%                |
| Storage Amount, Capacity          | 300 GB                                                                                                    | 250 GB             |
| Storage Amount, P95th Utilization | 1%                                                                                                        | 1.2%               |
| COST IMPACT                       |                                                                                                           |                    |
|                                   | CURRENT                                                                                                   | AFTER ACTIONS      |
| Compute Cost                      | \$146.91/mo                                                                                               | \$14.69/mo         |
| Storage Cost                      | \$11.05/mo                                                                                                | \$0.00/mo          |
| <b>Total Cost</b>                 | <b>\$157.96/mo</b>                                                                                        | <b>\$14.69/mo</b>  |
| <b>Total Savings</b>              |                                                                                                           | <b>\$143.27/mo</b> |

## Azure SQL DTU Databases

### Cost Calculation

For Azure SQL DTU Databases, the calculation for Estimated On-demand Monthly Cost can be expressed as follows:

$$(\text{On-demand Compute Rate} * 730) + (\text{Provisioned Database Storage Rate} * (\text{Provisioned Database Storage Amount} - \text{Performance Level Included Storage})) = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Compute Rate** is the **hourly** cost for a Database's instance type  
You can obtain on-demand rates via [Azure SQL Database Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.
- **Provisioned Database Storage Rate** is the cost for 1 GB / mo. of a Database's provisioned storage  
You can obtain provisioned database storage rates via [Azure SQL Database Pricing](#).
- **Performance Level Included Storage** is the storage amount included in the price of the selected Performance Level of a DTU Database

You can obtain information on DTU storage limits via [DTU Storage Limits](#).

The listed items above impact cost calculations and the scaling decisions that Turbonomic makes. These decisions also rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

### Example

Assume the following data for a pending scale action for an Azure SQL DTU Database:

|                                     | Current Values       | Values After Action Execution |
|-------------------------------------|----------------------|-------------------------------|
| On-demand Compute Rate              | \$0.20125/hr         | \$0.020125/hr                 |
| Provisioned Database Storage Rate   | \$0.221 per 1 GB/Mo. | \$0.221 per 1 GB/Mo.          |
| Performance Level Included Storage  | 250 GB               | 250 GB                        |
| Provisioned Database Storage Amount | 300 GB               | 250 GB                        |

Turbonomic calculates the following:

- *Current* Estimated On-demand Monthly Cost:

$$(\$0.20125 * 730) + (\$0.221 * (300 - 250)) = \$157.96/\text{Mo.}$$

- Estimated On-demand Monthly Cost *after* executing the action:

$$(\$0.020125 * 730) + (\$0.221 * (250 - 250)) = \$14.69/\text{Mo.}$$

### Note:

Turbonomic rounds the calculated values that it displays in the user interface.

The Estimated On-demand Monthly Cost is projected to decrease from \$157.96/month to \$14.69/month, as shown in the Details section of the pending action.

| COST IMPACT ?  | CURRENT     | AFTER ACTIONS |
|----------------|-------------|---------------|
| Compute Cost ⓘ | \$146.91/mo | \$14.69/mo    |
| Storage Cost ⓘ | \$11.05/mo  | \$0.00/mo     |
| Total Cost     | \$157.96/mo | \$14.69/mo    |
| Total Savings  |             | \$143.27/mo   |

Turbonomic treats the action as a saving, and shows an estimated savings of \$143.27/month.

| COST IMPACT ?  | CURRENT     | AFTER ACTIONS |
|----------------|-------------|---------------|
| Compute Cost ⓘ | \$146.91/mo | \$14.69/mo    |
| Storage Cost ⓘ | \$11.05/mo  | \$0.00/mo     |
| Total Cost     | \$157.96/mo | \$14.69/mo    |
| Total Savings  |             | \$143.27/mo   |

## Azure SQL vCore Databases

### Cost Calculation

For Azure SQL vCore Databases, the calculation for Estimated On-demand Monthly Cost can be expressed as follows:

$$(\text{On-demand Compute Rate} * 730) + (\text{SQL License Rate} * 730) + (\text{Provisioned Database Storage Rate} * (\text{Provisioned Database Storage Amount} + \text{Log Space Allocated})) = \text{Estimated On-demand Monthly Cost}$$

Where:

- **On-demand Compute Rate** is the hourly cost for a Database's instance type  
You can obtain on-demand rates via [Azure SQL Database Pricing](#).
- **730** represents the number of hours per month that Turbonomic uses to estimate monthly costs.
- **SQL License Rate** is the hourly cost for a Database's SQL license  
You can obtain SQL license rates via [Azure SQL Database Pricing](#).  
**Note:** "Pay as you go" prices in the link above represent the sum of compute and license costs, while "Azure Hybrid Benefit Price" values represent compute costs only.
- **Provisioned Database Storage Rate** is the cost for 1 GB / mo. of a Database's provisioned storage  
You can obtain provisioned database storage rates via [Azure SQL Database Pricing](#).
- **Log Space Allocated** is the log storage space automatically allocated to single Database instance by Azure.  
**Note:** Log storage space is considered in database cost calculations, but not reflected in Storage capacity.  
You can obtain provisioned database storage rates via [Azure SQL Database Pricing](#).

The listed items above impact cost calculations and the scaling decisions that Turbonomic makes. These decisions also rely on other factors, such as resource utilization percentiles and scaling constraints set in policies.

### Example

Assume the following data for a pending scale action for an Azure SQL vCore Database:

|                                     | Current Values | Values After Action Execution |
|-------------------------------------|----------------|-------------------------------|
| On-demand Compute Rate              | \$1.068/hr     | \$0.304/hr                    |
| SQL License Rate                    | \$0.799728/hr  | \$0.199932/hr                 |
| Provisioned Database Storage Rate   | \$0.115/hr     | \$0.115/hr                    |
| Provisioned Database Storage Amount | 32 GB          | 5 GB                          |

Turbonomic calculates the following:

- *Current* Estimated On-demand Monthly Cost:

$$(\$1.068 * 730) + (\$0.799728 * 730) + (\$0.115 * (32 + 9.6)) = \$1368.23/\text{Mo.}$$

- Estimated On-demand Monthly Cost *after* executing the action:

$$(\$0.304 * 730) + (\$0.199932 * 730) + (\$0.115 * (5 + 1.5)) = \$368.62/\text{Mo.}$$

### Note:

Turbonomic rounds the calculated values that it displays in the user interface.

Since the Estimated On-demand Monthly Cost is projected to decrease from \$1368.23/month to \$368.62/month, Turbonomic treats the action as a cost-saving measure and shows estimated savings of \$999.61/month.

## Cloud Database Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

For details about cloud database actions, see [Cloud Database Actions](#).

| Action                   | Default Setting |
|--------------------------|-----------------|
| Cloud DB Scale           | Manual          |
| Database Suspend Actions | On, Manual      |

### Scaling Sensitivity

Turbonomic uses a percentile of utilization over the specified observation period. This gives sustained utilization and ignores short-lived bursts.

Turbonomic uses these settings to calculate utilization percentiles for DTU and storage. It then recommends actions to improve utilization based on the observed values for a given time period.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 95th percentile. The percentile utilization is the highest value that 95% of the observed samples fall below. Compare that to average utilization, which is the average of *all* the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. This is important in the cloud, so that analysis can better exploit the elasticity of the cloud. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce capacity. Without using a percentile, Turbonomic never resizes below the recognized peak utilization. Assume utilization peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce resources for that database.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single burst to 100%, but for 95% of the samples, utilization never exceeded 50%. If you set **Aggressiveness** to 95th Percentile, then Turbonomic can see this as an opportunity to reduce resource allocation.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 99th Percentile – More performance. Recommended for critical databases that need maximum guaranteed performance at all times, or those that need to tolerate sudden and previously unseen spikes in utilization, even though sustained utilization is low.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings. This assures performance while avoiding reactive peak sizing due to transient spikes, thus allowing you to take advantage of the elastic ability of the cloud.
- 90th Percentile – More efficiency. Recommended for databases that can stand higher resource utilization.

By default, Turbonomic uses samples from the last 14 days. Use the **Max Observation Period** setting to adjust the number of days.

- **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 14 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the database has fewer days' data then it uses all of the stored historical data.

You can make the following settings:

- Less Elastic – Last 30 Days
- Recommended – Last 14 Days
- More Elastic – Last 7 Days or Last 3 Days

Turbonomic recommends an observation period of 14 days so it can recommend scaling actions more often. Since Azure SQL DB scaling is minimally disruptive, with near-zero downtime, scaling often should not introduce any noticeable performance risks.

**Note:**

For more information about Azure scaling downtimes, see the [Azure documentation](#).

• **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

This setting ensures historical data for a minimum number of days before Turbonomic will generate an action based on the percentile set in **Aggressiveness**. This ensures a minimum set of data points before it generates the action.

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

- More Elastic – None
- Less Elastic – 7 Days

## Cloud Instance Types

By default, Turbonomic considers all instance types currently available for scaling when making scaling decisions for databases. However, you may have set up your cloud databases to *only scale to* or *avoid* certain instance types to reduce complexity and cost, or meet demand. Use this setting to identify the instance types that databases can scale to.

| Attribute            | Default Value |
|----------------------|---------------|
| Cloud Instance Types | None          |

Click **Edit** to set your preferences. In the new page that displays, expand a **cloud tier** (a family of instance types, such as *Premium*) to see individual instance types and the resources allocated to them.

Select your preferred instance types or cloud tiers, or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

**Note:**

This policy setting is not available in plans.

If you selected a cloud tier and the service provider deploys new instance types to that tier later, then those instance types will automatically be included in your policy. Be sure to review your policies periodically to see if new instance types have been added to a tier. If you do not want to scale to those instance types, update the affected policies.

## Scaling Target Utilization

The utilization that you set here specifies the percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

The settings you make depend on the pricing model in place for the workloads in the policy scope. To meet a target DTU utilization, the workloads must be members of a DTU pricing model. To meet individual VCPU, VMEM, or IOPs/Throughput targets, the workloads must be members of a vCore pricing model.

| Attribute                      | DTU Pricing: Default Value | vCore Pricing: Default Value |
|--------------------------------|----------------------------|------------------------------|
| Scaling Target DTU Utilization | 70                         | N/A                          |
| VCPU                           | N/A                        | 70                           |
| VMEM                           | N/A                        | 90                           |
| IOPs/Throughput                | N/A                        | 70                           |
| Storage Amount Utilization     | 90                         | 90                           |

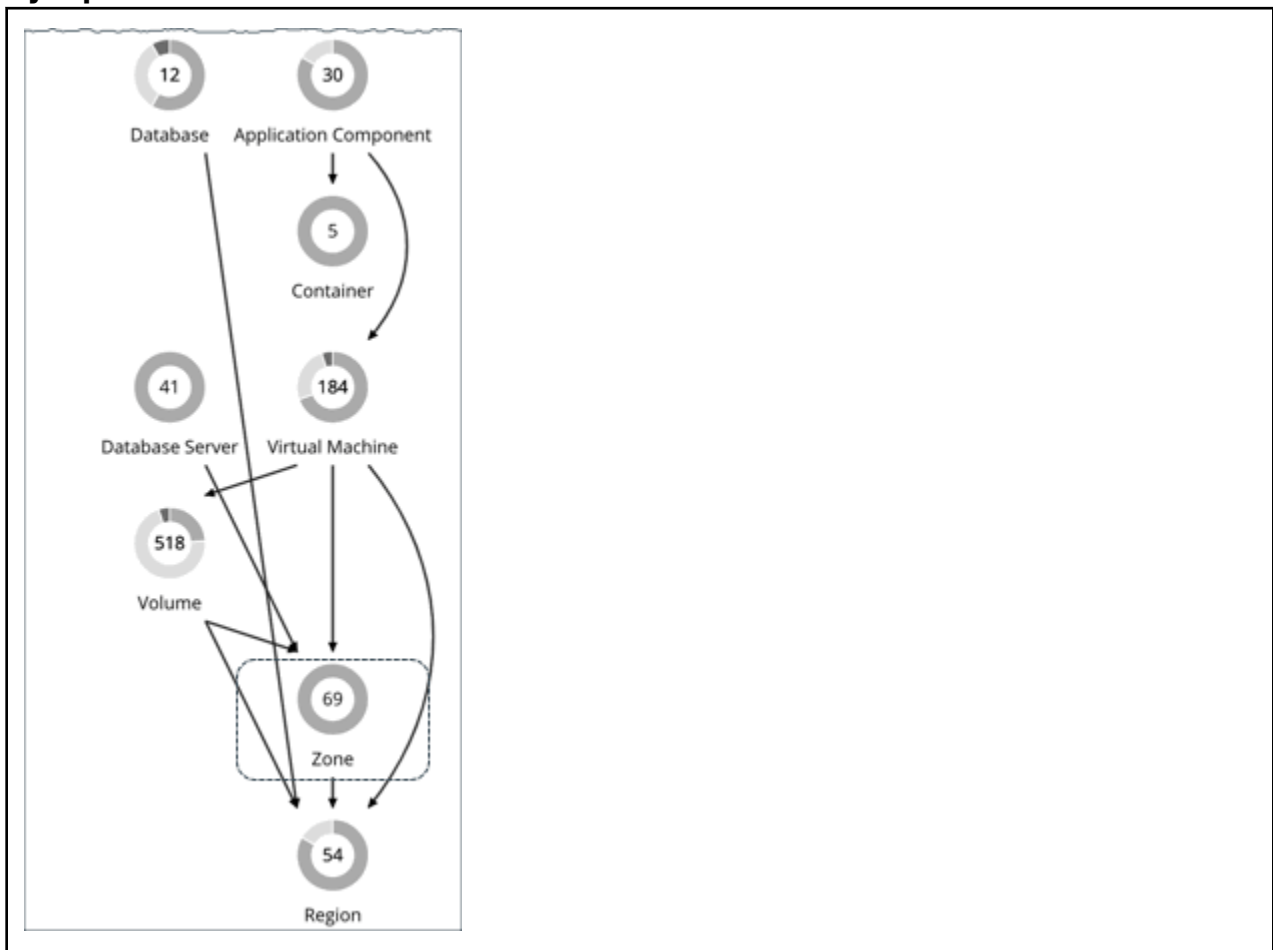
These advanced settings determine how much you would like a scope of workloads to utilize their resources. These are fixed settings that override the way Turbonomic calculates the optimal utilization of resources. You should only change these settings after consulting with Technical Support.

While these settings offer a way to modify how Turbonomic recommends actions, in most cases you should never need to use them. If you want to control how Turbonomic recommends actions to resize workloads, you can set the aggressiveness per the percentile of utilization, and set the length of the sample period for more or less elasticity on the cloud.

## Zone

A Zone represents an Availability Zone in your public cloud account or subscription. A zone is a location within a given region that serves as a datacenter to host the workloads that you run in your environment.

### Synopsis





| <b>Synopsis</b>     |                                                          |
|---------------------|----------------------------------------------------------|
| Budget:             | Turbonomic assumes a Zone has infinite resources.        |
| Provides:           | Compute and storage resources to VMs.                    |
| Consumes:           | Region resources.                                        |
| Discovered through: | Turbonomic discovers Zones through public cloud targets. |

**Monitored Resources**

For public cloud environments, Turbonomic discovers the resources that an availability zone provides, including:

- **Templates**  
The templates and template families that each zone or region delivers. This includes template capacity and cost for workload resources.
- **Account Services**  
These include storage modes, services the accounts offer for enhanced metrics, and services for different storage capabilities.
- **Relational Database Services (RDS)**  
The RDS capabilities each cloud account provides.
- **Storage Tiers**  
Turbonomic discovers the storage tier that supports your workloads, and uses the tier pricing to calculate storage cost.
- **Billing**  
Turbonomic discovers the billing across the zones and regions to predict costs in the future, and to track ongoing costs. This includes comparing on-demand pricing to discount billing.

Turbonomic monitors the following resources for a Zone:

- **Virtual Memory**  
The percentage utilized of memory capacity for all the workloads in the zone.
- **Virtual CPU**  
The percentage utilized of VCPU capacity for all the workloads in the zone.
- **Storage Access**  
For environments that measure storage access, the percentage utilized of access capacity for the zone.
- **Storage Amount**  
The percentage utilized of storage capacity for the zone.
- **IO Throughput**  
For environments that measure IO throughput, the percentage utilized of throughput capacity for the zone.
- **IO Throughput Read**  
For environments that measure IO throughput read, the percentage utilized of throughput capacity for the zone.
- **IO Throughput Write**

For environments that measure IO throughput write, the percentage utilized of throughput capacity for the zone.

- Net Throughput

For environments that measure Net throughput, the percentage utilized of throughput capacity for the zone.

- Net Throughput Inbound

For environments that measure Net throughput Inbound, the percentage utilized of throughput inbound capacity for the zone.

- Net Throughput Outbound

For environments that measure Net throughput Outbound, the percentage utilized of throughput outbound capacity for the zone.

## Actions

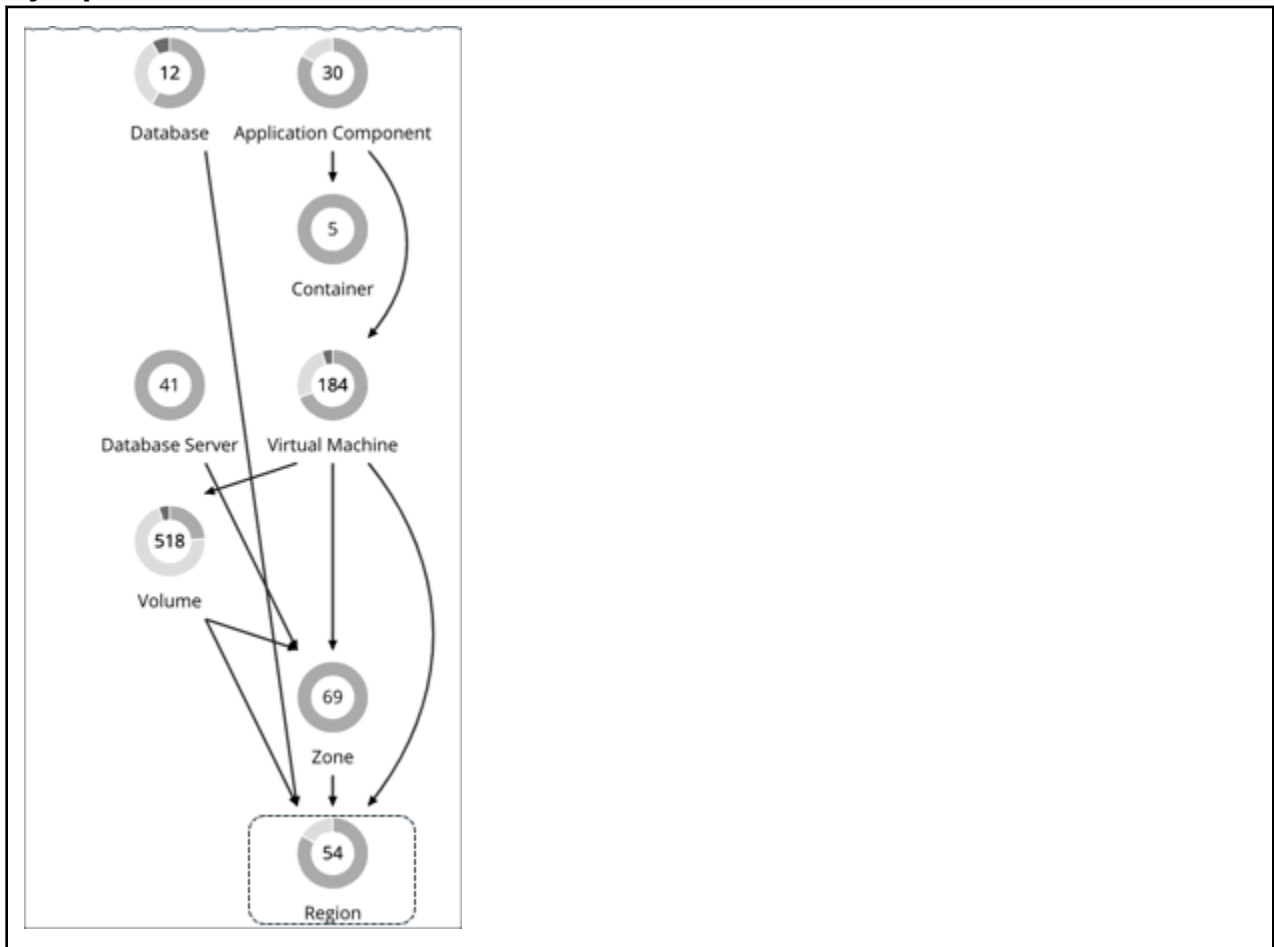
### None

Turbonomic does not recommend actions for a cloud zone.

## Region

A Region represents a geographical area that is home to one or more Availability Zones. Regions are often isolated from each other, and you can incur a cost for data transfer between them.

## Synopsis



| <b>Synopsis</b>     |                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------|
| Budget:             | Turbonomic assumes a Region has infinite resources.                                          |
| Provides:           | Hosting and storage resources to Zones.                                                      |
| Consumes:           | NA                                                                                           |
| Discovered through: | Cloud service accounts, such as accounts on Amazon AWS, or subscriptions on Microsoft Azure. |

## Monitored Resources

Turbonomic does not monitor resources directly from the region, but it does monitor the following resources, aggregated for the Zones in a region:

- Virtual Memory
  - The percentage utilized of memory capacity for workloads in the zones.
- Virtual CPU
  - The percentage utilized of VCPU capacity for workloads in the zones.
- Storage Access
  - For environments that measure storage access, the percentage utilized of access capacity for the zones.
- Storage Amount
  - The percentage utilized of storage capacity for the zones.
- IO Throughput
  - For environments that measure IO throughput, the percentage utilized of throughput capacity for the zones.
- IO Throughput Read
  - For environments that measure IO throughput read, the percentage utilized of throughput capacity for the zones.
- IO Throughput Write
  - For environments that measure IO throughput write, the percentage utilized of throughput capacity for the zones.
- Net Throughput
  - For environments that measure Net throughput, the percentage utilized of throughput capacity for the zones.
- Net Throughput Inbound
  - For environments that measure Net throughput Inbound, the percentage utilized of throughput inbound capacity for the zones.
- Net Throughput
  - For environments that measure Net throughput Outbound, the percentage utilized of throughput outbound capacity for the zones.

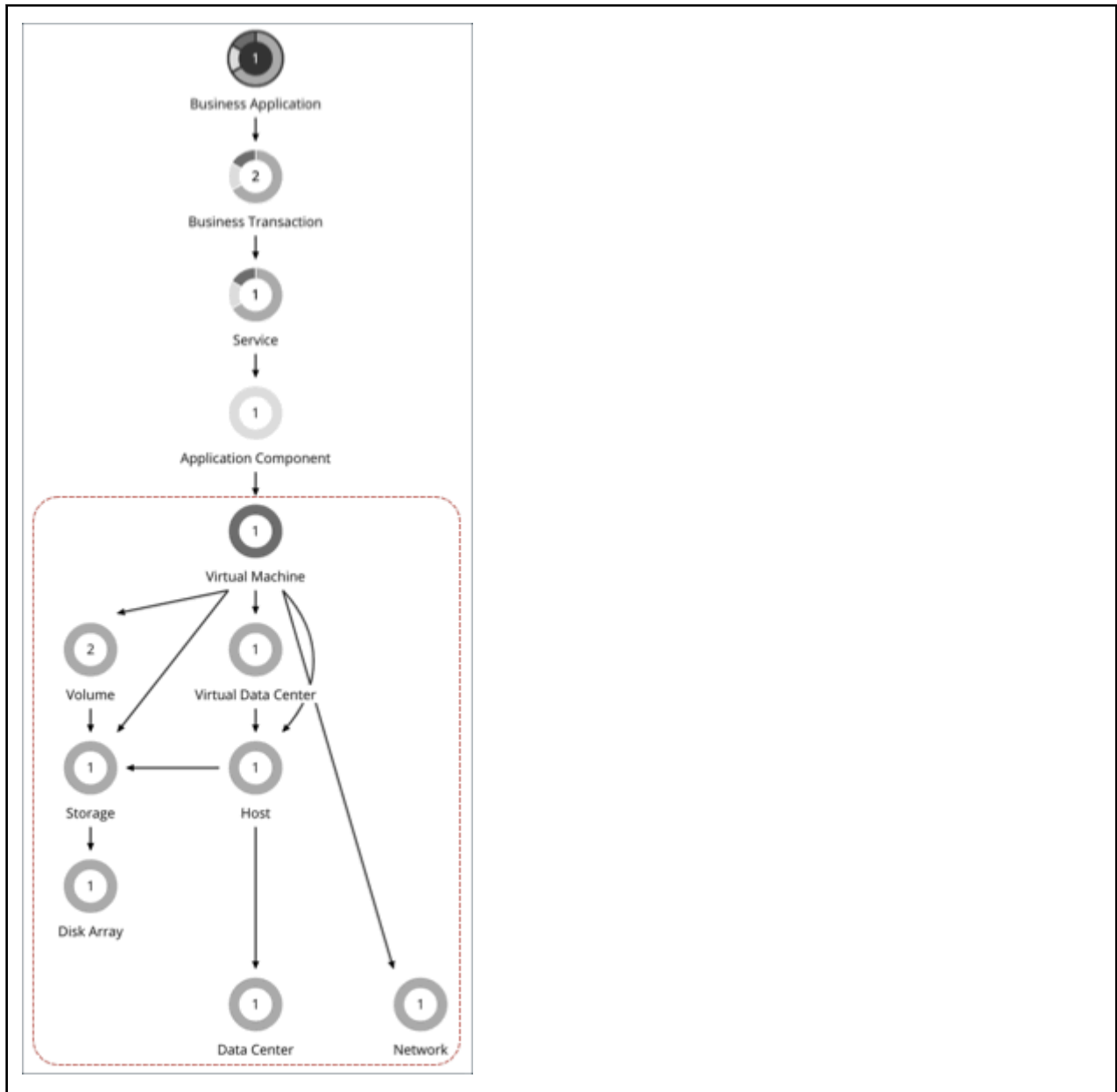
## Actions

### None

Turbonomic does not recommend actions for a cloud region.

## Entity Types - On-prem Infrastructure

Turbonomic discovers and monitors the entities that make up your on-prem infrastructure, and recommends actions to assure performance for the applications that consume resources from these entities.



### Virtual Machine (On-prem)

A virtual machine (VM) is a software emulation of a physical machine, including OS, virtual memory and CPUs, and network ports. VMs host applications, or they provide resources to container platforms.

**Note:**

Kubernetes nodes are represented as Virtual Machines in the Turbonomic supply chain. For details about nodes, see [Virtual Machine \(Kubernetes Node\)](#).



## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Provides:           | Resources for hosted applications to use: <ul style="list-style-type: none"><li>• VMEM</li><li>• VCPU</li><li>• Processing Units (IBM PowerVM)</li><li>• Virtual Processors (IBM PowerVM)</li><li>• VStorage</li><li>• IOPS (storage access operations per second)</li><li>• Latency (capacity for disk latency in ms)</li><li>• Memory and CPU Requests (for Kubernetes environments)</li></ul> |
| Consumes:           | <ul style="list-style-type: none"><li>• Physical host resources, including CPU and Mem</li><li>• Storage</li></ul>                                                                                                                                                                                                                                                                               |
| Discovered through: | Hypervisor targets                                                                                                                                                                                                                                                                                                                                                                               |

## Monitored Resources

Turbonomic monitors the following resources for a VM:

- Virtual Memory

Virtual Memory is the measurement of memory utilized by the entity.

- Virtual CPU

Virtual CPU is the measurement of CPU utilized by the entity.

- Processing Units (PU)

For PowerVM, processing units are a unit of measure for shared processing power across one or more virtual processors. One shared processing unit on one virtual processor accomplishes approximately the same work as one dedicated processor.

- Virtual Processors (VP)

For PowerVM, virtual processors (VPs) are the guest OS representation of the capacity of the VM. VPs have a one to one correlation with PUs, meaning 1 VP being utilized in a VM equates to 1 PU being used; however, VPs do not have to equal to PUs on a system. VPs are a virtual representation, therefore they can be over-provisioned.

- Virtual Storage (VStorage)

Virtual Storage is the measurement storage utilized by the entity.

- Storage Access

Storage Access is the measurement of IOPS utilized by the entity.

- Latency

Latency is the measurement of storage latency utilized by the entity.

For VMs discovered via vCenter, Cisco UCS, and HPE OneView, the following resources are also monitored:

- Energy

Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

- Carbon Footprint

Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

## Actions

### • Resize

When multiple resize actions that relate to the same vCenter VM are accepted (manually or automated), they execute together to avoid multiple restarts for the same VM. For example, a VM has both a vCPU resize action and a vMEM resize action. When selecting both actions and clicking Execute in the UI, the actions combine into the fewest vCenter tasks to minimize the disruption on the VM instead of one set of tasks per action. Only the VMware vCenter integration supports this functionality at this time.

#### – Resize resource capacity

Change the capacity of a resource that is allocated for the VM. For example, a resize action might recommend increasing the VMem available to a VM. Before recommending this action, Turbonomic verifies that the VM's cluster can adequately support the new size. If the cluster is highly utilized, Turbonomic will recommend a move action, taking into consideration the capacity of the new cluster and compliance with existing placement policies.

For hypervisor targets, Turbonomic can resize vCPU by changing the VM's socket or cores per socket count. For details, see [vCPU Scaling Controls](#).

#### – Resize resource reservation

Change the amount of a resource that is reserved for a VM. For example, a VM could have an excess amount of memory reserved. That can cause memory congestion on the host — A resize action might recommend reducing the amount reserved, freeing up that resource and reducing congestion

#### – Resize resource limit

Change the limit that is set on the VM for a resource. For example, a VM could have a memory limit set on it. If the VM is experiencing memory shortage, an action that decreases or removes the limit could improve performance on that VM.

### • Resize PU / Resize VP (IBM PowerVM)

Turbonomic does not support native execution of Resize PU (processing units) and Resize VP (virtual processor) actions. However, actions can be executed using external automation platforms through Action Orchestration workflows such as Action Scripts or Webhooks.

#### – Min/Max Observation Period

Observation period settings only apply to Resize VP actions.

For details, see [Aggressiveness and Observation Periods](#).

#### – Increment Constant for Processing Unit

Change the increment constant value to use for VM PU resize actions. PUs can be a decimal value in a VM setting.

For details, see [Increment Constant for Processing Unit](#).

#### – Increment Constant for Virtual Processors

Change the increment constant value to use for VM VP resize actions. VPs must be an integer value in a VM setting.

For details, see [Increment Constant for Virtual Processors](#).

### • Move

Move a VM due to:

- High resource utilization on VM or host
- Excess IOPS or latency in VStorage
- Workload placement violation
- Underutilized host (move VM before suspending host)

### • Reconfigure

Change a VM's configuration to comply with a policy.

For hypervisor targets, Turbonomic can reconfigure VMs that violate vCPU scaling policies. For details, see [VCPU Scaling Controls](#).

## Tuned Scaling for On-prem VMs

For resizing VMs, Turbonomic includes tuned scaling action settings. These settings give you increased control over the action acceptance mode for various resize actions. With this feature, you can automate resize actions within a normal range (the tuned scaling range), and direct Turbonomic to take more conservative actions when resizes are outside the range.

For example, consider resizing VMs to add more memory. As memory demand increases on a VM, Turbonomic can automatically allocate more memory. If the hosted application is in a runaway state (always requesting more memory) and ultimately falls outside of the normal range, Turbonomic will not automate memory resize for the VM.

To configure tuned scaling:

1. Create a VM policy.
2. Under **Action Automation**, configure the action acceptance mode for the various resize actions.
  - VCPU Resize Up
  - VCPU Resize Down
  - VCPU Resize Above Max
  - VCPU Resize Below Min
  - VMEM Resize Up
  - VMEM Resize Down
  - VMEM Resize Above Max
  - VMEM Resize Below Min

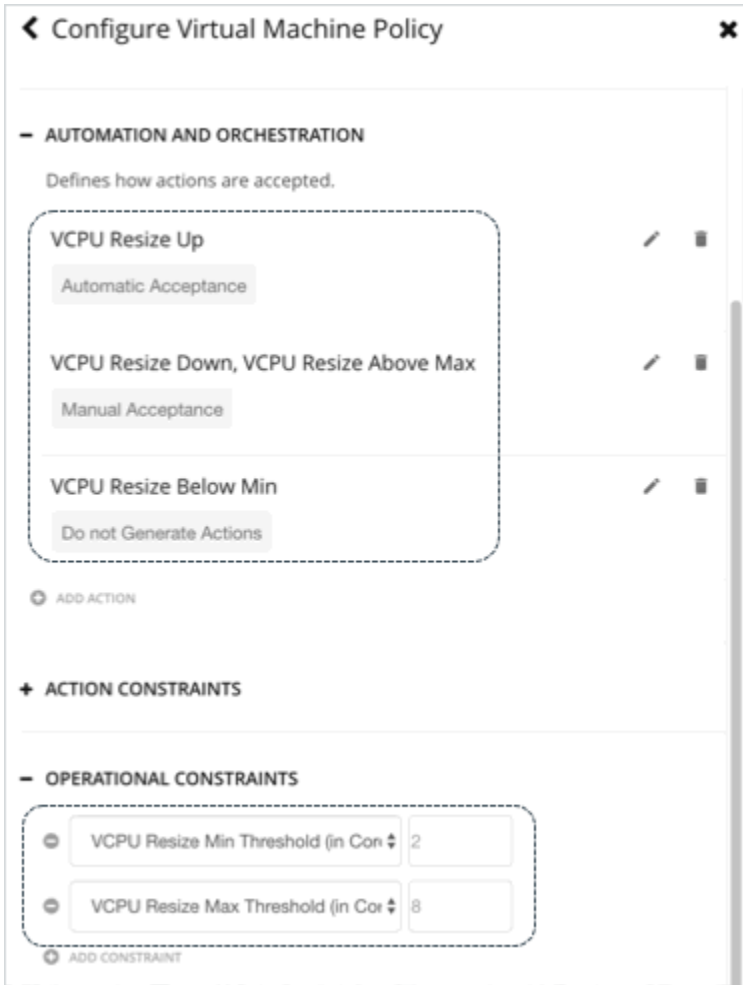
**Note:**

**Resize Up** and **Resize Down** settings are for conditions within the tuned scaling range, while **Above Max** and **Below Min** settings are for outlying conditions.

3. Under **Operational Constraints**, specify the tuned scaling range.
  - VCPU Resize Max Threshold
  - VCPU Resize Min Threshold
  - VMEM Resize Max Threshold
  - VMEM Resize Min Threshold

For example, assume the following settings:





As VCPU utilization for a VM changes over time, Turbonomic handles resize actions as follows.

| Current | Resize Request        | Action Acceptance Mode                                                                                                                                                                   | Result                                        |
|---------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| 6 VCPUs | Resize up to 8 VCPUs  | Automatic<br>Since the VM will have 8 VCPUs after the requested resize, which is within the VCPU Resize Max threshold of 8, Turbonomic executes the VCPU Resize Up action automatically. | 8 VCPUs                                       |
| 8 VCPUs | Resize up to 10 VCPUs | Manual<br>Since the VM will have 10 VCPUs after the requested resize, which is above the VCPU Resize Max threshold of 8, Turbonomic posts the VCPU Resize Up action (as a pending        | 10 VCPUs (if you executed the pending action) |

| Current  | Resize Request         | Action Acceptance Mode                                                                                                                                                                                                                                                 | Result                                       |
|----------|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
|          |                        | action) and provides the option to execute that action through the user interface.                                                                                                                                                                                     |                                              |
| 10 VCPUs | Resize down to 2 VCPUs | Manual<br>Since the VM will have 2 VCPUs after the requested resize, which is within the VCPU Resize Min threshold of 2, Turbonomic posts the VCPU Resize Down action (as a pending action) and provides the option to execute that action through the user interface. | 2 VCPUs (if you executed the pending action) |
| 2 VCPUs  | Resize down to 1 VCPU  | Not Generated<br>Since the VM will have 1 VCPU after the requested resize, which is below the VCPU Resize Min threshold of 2, Turbonomic does not generate the VCPU Resize Down action to comply with the policy.                                                      | 2 VCPUs                                      |

Action policies include scope to determine which entities will be affected by the given policy. It's possible for two or more policies to affect the same entities. As is true for other policy settings, tuned scaling uses the most conservative settings for the affected entities. The effective action acceptance mode will be the most conservative, and the effective tuned scaling range will be the narrowest range (the lowest MAX and highest MIN) out of the multiple policies that affect the given entities. For more information, see [Default and User-defined Policies](#).







You can schedule automation policies to take effect during a certain window of time. You can include tuned scaling settings in a scheduled window, the same as you can schedule other policy settings. For more information, see [Policy Schedule](#).





## On-prem VM Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about on-prem VM actions, see [On-prem VM Actions](#).

| Action                               | Default Mode | vCenter                                                                             | Hyper-V                                                                                                                                                                                             | PowerVM                                                                               |
|--------------------------------------|--------------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Move                                 | Manual       |    |                                                                                                                   | N/A                                                                                   |
| Reconfigure                          | Recommend    |    |                                                                                                                   | N/A                                                                                   |
| Start                                | Manual       |    |                                                                                                                   | N/A                                                                                   |
| Storage Move                         | Recommend    |    | With VMM:<br><br>Otherwise:<br> | N/A                                                                                   |
| Provision<br>(Kubernetes nodes only) | Manual       |    |                                                                                                                   | N/A                                                                                   |
| Suspend<br>(Kubernetes nodes only)   | Manual       |    |                                                                                                                   | N/A                                                                                   |
| PU Resize Up                         | Recommend    | N/A                                                                                 | N/A                                                                                                                                                                                                 |  |
| PU Resize Down                       | Recommend    | N/A                                                                                 | N/A                                                                                                                                                                                                 |  |
| VP Resize Up                         | Recommend    | N/A                                                                                 | N/A                                                                                                                                                                                                 |  |
| VP Resize Down                       | Recommend    | N/A                                                                                 | N/A                                                                                                                                                                                                 |  |
| vCPU Resize Up*                      | Manual       |  |                                                                                                                 | N/A                                                                                   |
| vCPU Resize Above Max*               | Recommend    |  |                                                                                                                 | N/A                                                                                   |
| vCPU Resize Down*                    | Manual       |  |                                                                                                                 | N/A                                                                                   |
| vCPU Resize Below Min*               | Recommend    |  |                                                                                                                 | N/A                                                                                   |
| vMem Resize Up*                      | Manual       |  |                                                                                                                 | N/A                                                                                   |
| vMem Resize Above Max*               | Recommend    |  |                                                                                                                 | N/A                                                                                   |

| Action                 | Default Mode | vCenter                                                                           | Hyper-V                                                                            | PowerVM |
|------------------------|--------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------|
| vMem Resize Down*      | Manual       |  |  | N/A     |
| vMem Resize Below Min* | Recommend    |  |  | N/A     |

\* *Turbonomic uses these settings, in conjunction with [resize operational constraints](#), to set up [tuned scaling](#) for on-prem VMs.*

You can use [Action Scripts](#) and third-party orchestrators (such as ServiceNow) for action orchestration.

## Non-disruptive Mode

VM actions include the modifier, **Enforce Non Disruptive Mode**. When you enable this modifier, Turbonomic ensures that a resize action in *Automatic* or *Manual* mode will not require a reboot or any other disruption to the affected VM. If the action will disrupt the VM, Turbonomic posts the action in *Recommend* mode.

| Attribute                   | Default Setting |
|-----------------------------|-----------------|
| Enforce Non-disruptive Mode | Off             |

This setting has no effect on actions set to *Recommend* mode. Turbonomic will continue to post those actions for you to evaluate.

You can enforce non disruptive mode in the default VM policy, and then schedule action policies to automate resize actions during downtimes. Be aware that scheduled actions do not respect the enforced non disruptive mode — Scheduled resize actions will execute during the scheduled window even if they require a reboot. This is useful for setting up certain action behaviors, but you must be aware that enforced non disruptive mode has no effect on scheduled actions.

### Note:

When you configure a schedule window for a VM resize action, to ensure Turbonomic will execute the action during the scheduled time, you must turn off the **Enforce Non Disruptive Mode** setting for that scheduled policy. Even if you turn the setting off for the global policy, you still must turn the setting off for your scheduled policy. Otherwise Turbonomic will not execute the resize action.

## Hypervisor VMEM for Resize

For on-prem environments, Turbonomic discovers VMEM utilization and can recommend actions to resize the VMEM capacity on a VM. For environments that do not include Applications and Databases as targets, the data that analysis uses to make these recommendations comes from the underlying hypervisors. However, that data is not always sufficient to result in accurate resize recommendations. Use the **Use Hypervisor VMEM for Resize** setting to determine how to generate VMEM recommendations.

| Attribute                  | Default Setting |
|----------------------------|-----------------|
| Hypervisor VMEM for Resize | On              |

- **On**

When your environment includes Applications and Databases as targets, Turbonomic uses the VMEM metrics those targets discover. If a scope of VMs does not fall under those targets, then analysis *will* generate VMEM resize actions for that scope. In this case, analysis uses the VMEM metrics it discovers from the underlying hypervisors.

- **Off**

When your environment includes Applications and Databases as targets, Turbonomic uses the VMEM metrics those targets discover. If a scope of VMs does not fall under those targets, then analysis *will not* generate VMEM resize actions for that scope.

## Shared Nothing Migration

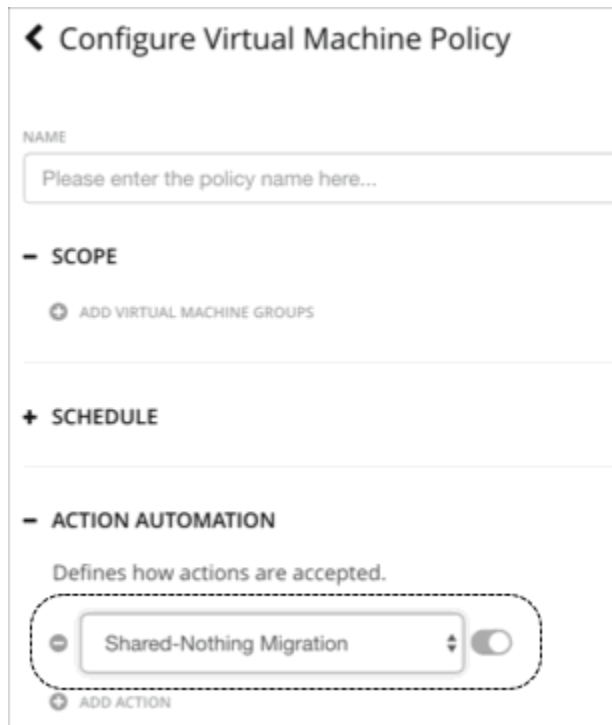
If you have enabled both storage and VM moves, Turbonomic can perform shared-nothing migrations, which move the VM and the stored VM files simultaneously. For example, assume a VM on a host also uses local storage on that host. In that case, Turbonomic can move that VM and move its data to a different datastore in a single action.

| Attribute                | Default Setting |
|--------------------------|-----------------|
| Shared-nothing Migration | Off             |

Currently, the following targets support shared-nothing migrations:

- vSphere, versions 5.1 or greater
- VMM for Hyper-V 2012 or later

Because of this feature's potential impact on performance, it is turned off by default. Turbonomic recommends enabling it only on VMs that need it. To do this, you must first set the action acceptance mode for VM and storage moves to either *Manual* or *Automatic*, and then enable the feature in a VM policy.



If a policy that enables this feature conflicts with a more conservative policy, the latter policy wins. For example, if compute move is set to *Manual*, storage move is set to *Recommend*, and shared-nothing migration is turned on, shared-nothing migration is in effect but remains in *Recommended* state.

**Note:**

Turbonomic does not simulate shared-nothing migrations in plans.

## Resize Thresholds (Operational Constraints)

Turbonomic uses these settings to set up **tuned scaling** actions for on-prem VMs. Tuned scaling gives you increased control over the action mode for various resize actions. With this feature, you can automate resize actions within a normal range (the tuned scaling range), and direct Turbonomic to take more conservative actions when resizes are outside the range.

For details about tuned scaling, see [Tuned Scaling for On-prem VMs](#).

| Attribute                            | Default Value                             |
|--------------------------------------|-------------------------------------------|
| VCPU Resize Max Threshold (in Cores) | 64<br>Tuned Scaling Range Upper Limit     |
| VCPU Resize Min Threshold (in Cores) | 1<br>Tuned Scaling Range Lower Limit      |
| VMEM Resize Max Threshold (MB)       | 131072<br>Tuned Scaling Range Upper Limit |
| VMEM Resize Min Threshold (MB)       | 512<br>Tuned Scaling Range Lower Limit    |

## Resize VStorage

The default setting disables resize actions. This is usually preferred because VStorage resize requires that you reformat the storage. The increment constant takes effect if you enable resizing.

| Attribute                            | Default Setting/Value                                                                                              |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Resize VStorage                      | Disabled                                                                                                           |
| Increment constant for VStorage [GB] | None<br>If you enable resize, Turbonomic uses the default value of 1024. You can change this to a different value. |

## vCPU Scaling Controls

For details, see [vCPU Scaling Controls](#).

## Resize Increments

These increments specify how many units to add or subtract when resizing the given resource allocation for a VM.

| Attribute                            | Default Value |
|--------------------------------------|---------------|
| Increment constant for VMEM [MB]     | 1024          |
| Increment constant for VStorage [GB] | 1024          |

**Note:**

vCPU resize increments are configured in conjunction with vCPU scaling controls. For details, see [VCPU Scaling Controls](#).

For VMem, you should not set the increment value to be lower than what is necessary for the VM to operate. If the VMem increment is too low, then it's possible that Turbonomic would allocate insufficient VMem for the machine to operate. For a VM that is under utilized, Turbonomic will reduce VMem allocation by the increment amount, but it will not leave a VM with zero VMem. For example, if you set this to 1024, then Turbonomic cannot reduce the VMem to less than 1024 MB.

## Increment Constant for Processing Units

For PowerVM, processing units are a unit of measure for shared processing power across one or more virtual processors. One shared processing unit on one virtual processor accomplishes approximately the same work as one dedicated processor.

Turbonomic recommends VM PU resize actions in multiples of this increment constant value. For example, if Turbonomic determines that a VM with 1.0 PU needs 1.2 PU and the increment is set to 0.5, then a resize up action is not generated. Increasing this increment reduces the frequency of resize actions so that when they appear the change is more significant.

| Attribute                               | Default Value |
|-----------------------------------------|---------------|
| Increment constant for processing units | 0.1           |

## Increment Constant for Virtual Processors

For PowerVM, virtual processors (VPs) are the guest OS representation of the capacity of the VM. VPs have a one to one correlation with PUs, meaning 1 VP being utilized in a VM equates to 1 PU being used; however, VPs do not have to equal to PUs on a system. VPs are a virtual representation, therefore they can be over-provisioned.

Turbonomic recommends VM VP resize actions in multiples of this increment constant value. For example, if Turbonomic determines that a VM with 2 VPs needs 1 VP and the increment is set to 2, then a resize down action is not generated. Increasing this increment reduces the frequency of resize actions so that when they appear the change is more significant.

| Attribute                                 | Default Value |
|-------------------------------------------|---------------|
| Increment constant for virtual processors | 1             |

## Aggressiveness and Observation Periods

Turbonomic uses these settings to calculate utilization percentiles for vCPU, vMEM, IOPS, and VPs. It then recommends actions to improve utilization based on the observed values for a given time period.

- **Aggressiveness**

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 75th Percentile |

When evaluating performance, Turbonomic considers resource utilization as a percentage of capacity. The utilization drives actions to scale the available capacity either up or down. To measure utilization, the analysis considers a given utilization percentile. For example, assume a 75th percentile. The percentile utilization is the highest value that 75% of the observed samples fall below. Compare that to average utilization, which is the average of all the observed samples.

Using a percentile, Turbonomic can recommend more relevant actions. For scheduled policies, the more relevant actions will tend to remain viable when their execution is put off to a later time.

For example, consider decisions to reduce the capacity for CPU on a VM. Without using a percentile, Turbonomic never resizes below the recognized peak utilization. For most VMs, there are moments when peak CPU reaches high levels, such as during reboots, patching, and other maintenance tasks. Assume utilization for a VM peaked at 100% just once. Without the benefit of a percentile, Turbonomic will not reduce allocated CPU for that VM.

With **Aggressiveness**, instead of using the single highest utilization value, Turbonomic uses the percentile you set. For the above example, assume a single CPU burst to 100%, but for 75% of the samples CPU never exceeded 50%. If you set **Aggressiveness** to 75th Percentile, then Turbonomic can see this as an opportunity to reduce CPU allocation for the VM.

In summary, a percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples.

By default, Turbonomic uses samples from the last 30 days. Use the **Max Observation Period** setting to adjust the number of days. To ensure that there are enough samples to analyze and drive scaling actions, set the **Min Observation Period**.

- **Max Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 90 Days  |

To refine the calculation of resource utilization percentiles, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. If the database has fewer days' data then it uses all of the stored historical data.

You can make the following settings:

- Less Elastic – Last 90 Days
- Recommended – Last 30 Days
- More Elastic – Last 7 Days

Turbonomic recommends an observation period of 30 days following the monthly workload maintenance cycle seen in many organizations. VMs typically peak during the maintenance window as patching and other maintenance tasks are carried out. A 30-day observation period means that Turbonomic can capture these peaks and increase the accuracy of its sizing recommendations.

You can set the value to 7 days if workloads need to resize more often in response to performance changes. For workloads that cannot handle changes very often or have longer usage periods, you can set the value to 90 days.

- **Min Observation Period**

| Attribute              | Default Value |
|------------------------|---------------|
| Min Observation Period | None          |

Especially for scheduled actions, it is important that resize calculations use enough historical data to generate actions that will remain viable even during a scheduled maintenance window. A maintenance window is usually set for "down" time, when utilization is low. If analysis uses enough historical data for an action, then the action is more likely to remain viable during the maintenance window.

- More Elastic – None
- Less Elastic – 7 Days



## Rate of Resize

When resizing resources for a VM, Turbonomic calculates the optimal values for VMem, VCPU, and VStorage. But it does not necessarily make a change to that value in one action. Turbonomic uses the Rate of Resize setting to determine how to make the change in a single action.

### Note:

In Turbonomic 8.9.1, the Rate of Resize default value changed from 2 to 3. If you have changed your default setting to 1 or want to keep the current default setting of 2, create a new policy scoped to all on-prem VMs and configure the Rate of Resize to your desired setting.

| Attribute      | Default Value |
|----------------|---------------|
| Rate of Resize | High (3)      |

#### • Low

Change the value by one increment, only. For example, if the resize action calls for increasing VMem, and the increment is set at 1024, Turbonomic increases VMem by 1024 MB.

#### • Medium

Change the value by an increment that is 1/4 of the difference between the current value and the optimal value. For example, if the current VMem is 2 GB and the optimal VMem is 10 GB, then Turbonomic will raise VMem to 4 GB (or as close to that as the increment constant will allow).

#### • High

Change the value to be the optimal value. For example, if the current VMem is 2 GB and the optimal VMem is 8 GB, then Turbonomic will raise VMem to 8 GB (or as close to that as the increment constant will allow).

## Consistent Resizing

| Attribute                  | Default Setting |
|----------------------------|-----------------|
| Enable Consistent Resizing | Off             |

When you create a policy for a group of VMs and turn on Consistent Resizing, Turbonomic resizes all the group members to the same size, such that they all support the top utilization of each resource commodity in the group. For example, if you have deployed load balancing for a group, then all the VMs in the group should experience similar utilization. In that case, if one VM needs to be resized, then it makes sense to resize them all consistently.

Assume VM A shows top utilization of CPU, and VM B shows top utilization of memory. A resize action would result in all the VMs with CPU capacity to satisfy VM A, and memory capacity to satisfy VM B.

### Note:

If the VMs in the group have different core speeds, then CPU scaling actions might not be consistent. For example, if you set the maximum target CPU size to 2, Turbonomic might recommend resizing to more than 2 CPUs to account for the VMs with slower cores.

To avoid this problem, be sure that the group only includes VMs with the same core speed.

For an affected resize, the Actions List shows individual resize actions for each of the VMs in the group. To avoid the possibility of resizing VMs disruptively at the same time, you must create automation policies with non-overlapping schedules. For example, if VMs A and B are in the same consistent resizing group, create two policies that resize the VMs at different times of the day.

- For Policy 1, set the scope to a group containing VM A and enable resize automation between, say, 01:00 and 01:45.

- For Policy 2 set the scope to a group containing VM B and enable resize automation between 02:00 and 02:45.

When working with Consistent Resizing, consider these points:

- You should not mix VMs in a group that has a Consistent Resizing policy, with other groups that enable Consistent Resizing. One VM can be a member of more than one group. If one VM (or more) in a group with Consistent Resizing is also in another group that has Consistent Resizing, then both groups enforce Consistent Resizing together, for all their group members.
- For any group of VMs that enables Consistent Resizing, you should not mix the associated target technologies. For example, one group should not include VMs that are on Hyper-V and vCenter platforms.
- Charts that show actions and risks assign the same risk statement to all the affected VMs. This can seem confusing. For example, assume one VM needs to resize to address vCPU risk, and 9 other VMs are set to resize consistently with it. Then charts will state that 10 VMs need to resize to address vCPU risks.

## Placement Policies

Turbonomic supports placement policies for on-prem VMs, as follows:

- You can create placement policies to enforce constraints for VM placements.

For example, the VMs in a consumer group can only run on a host that is in the provider group. You can limit the number of consumers that can run on a single provider — for hosts in the provider group, only 2 instances of VMs in the consumer group can run on the same host. Or no more than the specified number of VMs can use the same storage device.

- For VMs that require paid licenses, you can create placement policies that set up certain hosts to be the VMs' preferred license providers. Turbonomic can then recommend consolidating VMs or reconfiguring hosts in response to changing demand for licenses.

For more information, see [Creating Placement Policies](#).

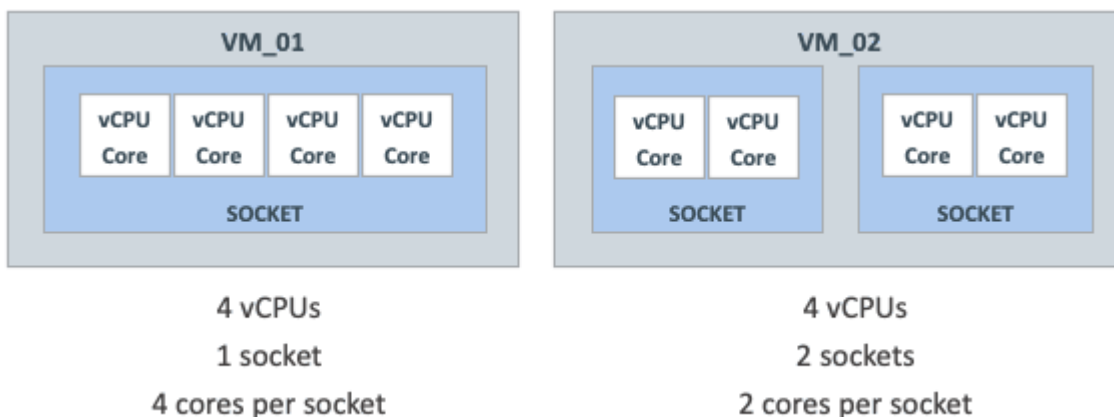
### Note:

For VMM targets, IBM automatically imports your Availability Sets, representing them as placement policies for the affected infrastructure. To see these availability sets, go to the **Settings > Policies** page and click **Imported Placement Policies**.

For more information, see [Importing Workload Placement Policies](#).

## vCPU Scaling Controls

Turbonomic represents the compute capacity of a VM in MHz and vCPUs. The following diagram shows how a VM with four vCPUs can be configured differently in terms of sockets and cores.



Turbonomic can resize the compute capacity by changing the number of sockets or cores per socket, depending on:

- The policy assigned to the VM

On-prem VM policies include vCPU Scaling Controls that give you granular control over how VM compute resources are *resized* to maintain performance or *reconfigured* to comply with your operational policies. You can create policies for different VM groups based on their resource needs and characteristics, and decide whether to automate resize and reconfigure actions in those policies.

- The hypervisor that manages the VM

Hypervisor targets have varying degrees of support for vCPU Scaling Controls. VMware vSphere supports all scaling controls, while Hyper-V and Nutanix AHV provide limited support. For details, see the *Hypervisor Support* section below.

## vCPU Scaling Control Modes and Options

Turbonomic provides **simple** and **advanced** controls to automate compute resource management actions in compliance with your policies. It also provides a **legacy** control based on units of MHz.

The controls you choose depend on your operational policies regarding the VM configuration of sockets and cores per socket, and your choice of hypervisor. For example, your operational policies may dictate a certain VM configuration that must be respected when resizing a VM's compute resources. Changing sockets is the least disruptive, but for some workloads, it may be preferable to change cores per socket due to socket licensing or operating system constraints. For larger VMs where Non-Uniform Memory Access (NUMA) must be considered for performance reasons, it may be preferable to balance vCPUs across host sockets.

The following tables explain the exact operation for each mode.

### Simple Controls

Simple controls change compute resources based on units of vCPU.

| vCPU Scaling Option | Unit  | Sockets            | Cores Per Socket                  | Resize Action                                                                                                                                                                                                    | Reconfigure Action |
|---------------------|-------|--------------------|-----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|
| Change virtual CPUs | vCPUs | Turbonomic decides | Reconfigured to 1 core per socket | <ul style="list-style-type: none"> <li>• Non-disruptive if hot-add is enabled and VM sockets are increasing</li> <li>• Disruptive if VM cores per socket does not equal 1, even if hot-add is enabled</li> </ul> | Disruptive         |

### Advanced Controls

Advanced controls allow you to change sockets or cores per socket, and configure additional options.

| <b>vCPU Scaling Option</b> | <b>Unit</b>       | <b>Sockets</b>            | <b>Cores Per Socket</b>            | <b>Resize Action</b>                                                                                                                                                      | <b>Reconfigure Action</b>                                                                                                                                                   |
|----------------------------|-------------------|---------------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Change sockets             | 1 socket          | Turbonomic decides        | Preserve VM cores per socket       | Non-disruptive if hot-add is enabled and VM sockets are increasing                                                                                                        | Not generated                                                                                                                                                               |
| Change sockets             | 1 socket          | Turbonomic decides        | User-specified VM cores per socket | <ul style="list-style-type: none"> <li>• Non-disruptive if hot-add is enabled</li> <li>• Disruptive if VM cores per socket does not match user-specified value</li> </ul> | Disruptive if VM cores per socket does not match user-specified value                                                                                                       |
| Change cores per socket    | 1 core per socket | Preserve VM sockets       | Turbonomic decides                 | Disruptive                                                                                                                                                                | Not generated                                                                                                                                                               |
| Change cores per socket    | 1 core per socket | Match host sockets        | Turbonomic decides                 | Disruptive                                                                                                                                                                | <ul style="list-style-type: none"> <li>• Non-disruptive if hot-add is enabled and VM sockets are increasing</li> <li>• Disruptive if cores per socket is changed</li> </ul> |
| Change cores per socket    | 1 core per socket | User-specified VM sockets | Turbonomic decides                 | Disruptive                                                                                                                                                                | <ul style="list-style-type: none"> <li>• Non-disruptive if hot-add is enabled and VM sockets are increasing</li> <li>• Disruptive if cores per socket is changed</li> </ul> |

### **Legacy Controls**

Legacy controls change compute resources based on units of MHz.

| <b>vCPU Scaling Option</b> | <b>Unit</b> | <b>Sockets</b>     | <b>Cores Per Socket</b>                                                                                                          | <b>Resize Action</b>                                               | <b>Reconfigure Action</b> |
|----------------------------|-------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|---------------------------|
| MHz legacy behavior        | MHz         | Turbonomic decides | <ul style="list-style-type: none"> <li>Assumes 1 core per socket</li> <li>Execution preserves actual cores per socket</li> </ul> | Non-disruptive if hot-add is enabled and VM sockets are increasing | Not generated             |

**Points to consider:**

- If non-disruptive mode is enabled, disruptive actions are not automated and must be executed manually.
- Older Guest OSes and applications may be sensitive to changes in the vCPU architecture that could result in power-on issues or kernel panics/BSODs. Some workloads require manual help with such changes, so always test certain classes of applications and guest operating systems before enabling any automation that changes the vCPU architecture. Use the Turbonomic knowledge of the application domain and Guest OS to scope them out of policies.

**Scaling Option: Change Virtual CPUs**

In this scaling option, Turbonomic adds or removes compute resources in increments of vCPUs. To achieve this, it changes the number of VM sockets and enforces 1 core per socket (if not already enforced).

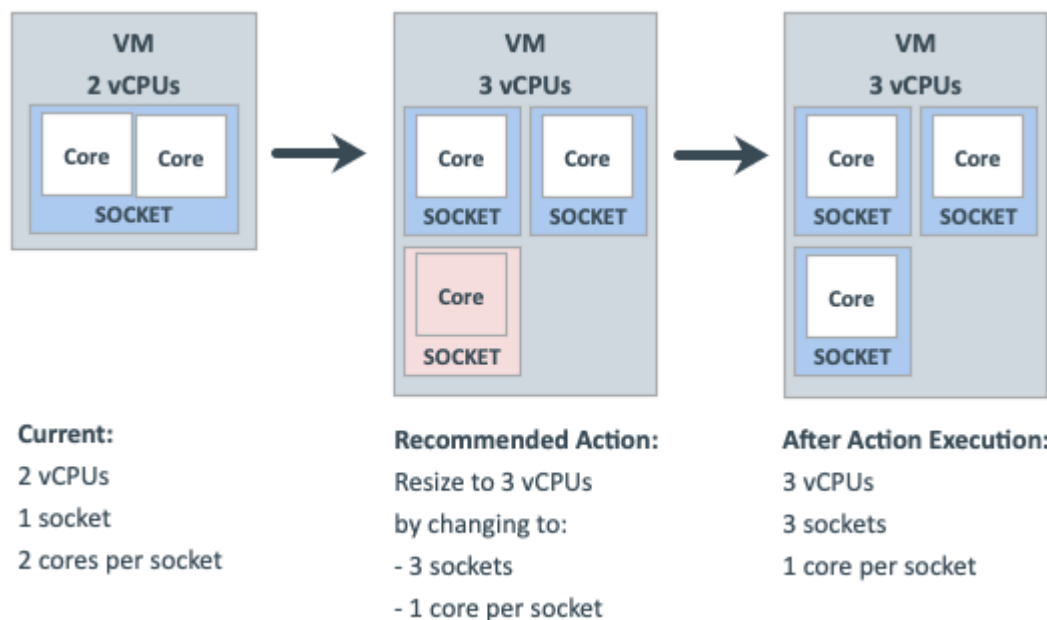
- If a VM requires a change to compute resources, Turbonomic generates a resize vCPU action that assumes 1 core per socket. If the VM currently does not have 1 core per socket, Turbonomic reconfigures it to 1 core per socket as part of action execution.
- If a VM is already optimally sized, but its current cores per socket is not 1, Turbonomic generates a reconfigure vCPU action to change cores per socket to 1, thereby bringing the VM into compliance with the policy.

This scaling option is ideal under the following scenarios:

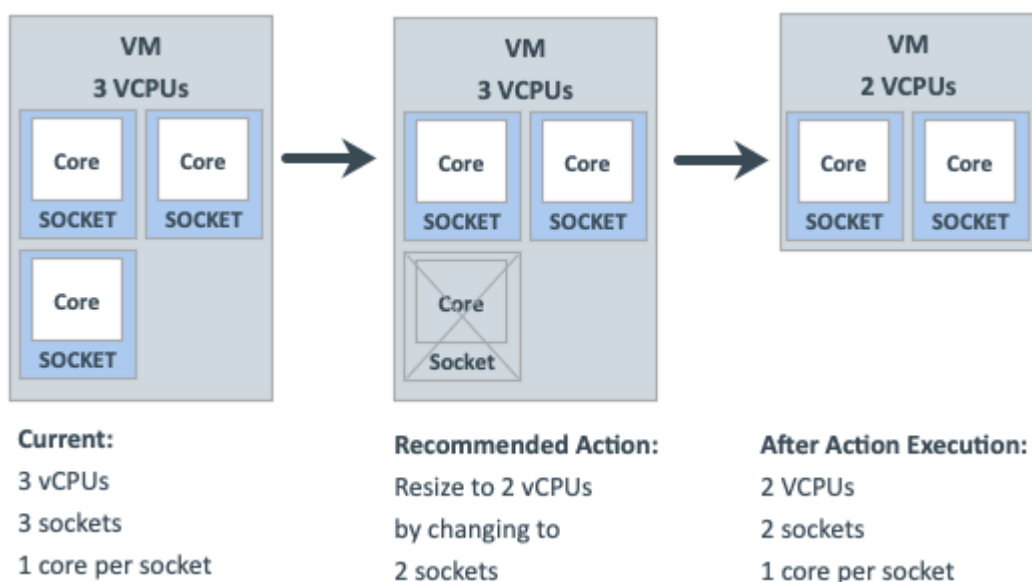
- Your environment has a large number of small VMs where precise vCPU scaling is the priority.
- You have VMs that already have 1 core per socket and require on-demand upsizes on these VMs to be non-disruptive.

For example, a VM currently has 1 socket and 2 cores per socket, and applies a policy that changes vCPU in increments of 1.

- If Turbonomic determines that the VM needs to increase compute capacity by 1 vCPU (i.e., from 2 to 3 vCPUs), a resize up action changes sockets from 2 to 3, and cores per socket from 2 to 1.



- When the same VM needs to reduce compute capacity by 1 vCPU (i.e., from 3 to 2 vCPUs), a resize down action changes sockets from 3 to 2.



### Scaling Option: Change Sockets

In this scaling option, Turbonomic adds or removes compute resources by changing VM sockets.

- If a VM requires a change to compute resources, Turbonomic generates a resize vCPU action that considers the current cores per socket value (if the 'Preserve existing VM cores per socket' option is set) or uses the user-specified cores per socket value. If the VM's current cores per socket value violates a policy (i.e., does not match the user-specified value), Turbonomic reconfigures the VM's cores per socket value as part of action execution, thereby bringing the VM into compliance with the policy, while at the same time providing the required change to compute resources.
- If a VM is already optimally sized, but its current cores per socket value violates a policy (i.e., does not match the user-specified value, if set), Turbonomic generates a reconfigure vCPU action to change cores per socket to the user-specified value, thereby bringing the VM into compliance with the policy.

#### Change Sockets and Preserve VM Cores Per Socket

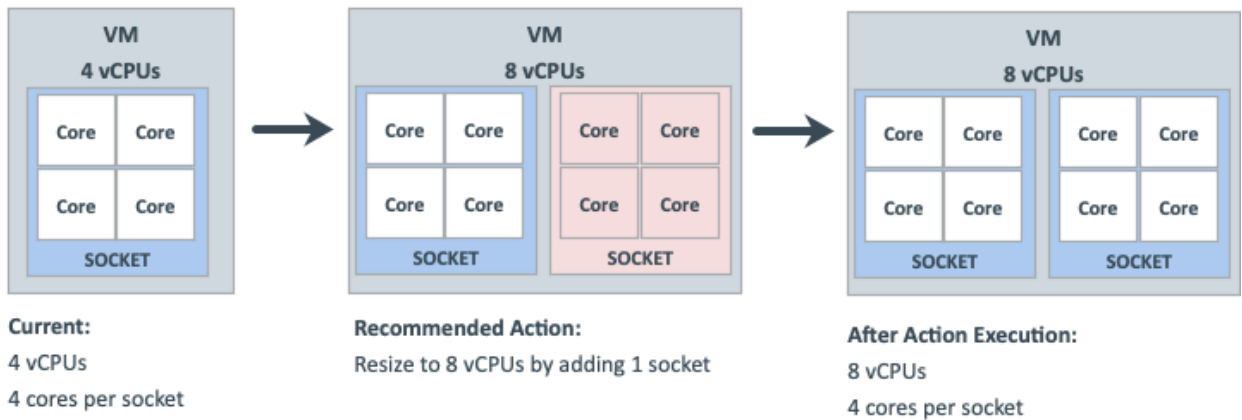
In this scaling option, Turbonomic adds or removes compute resources by changing VM sockets in increments of 1, and preserves VM cores per socket.

This scaling option is ideal under the following scenarios:

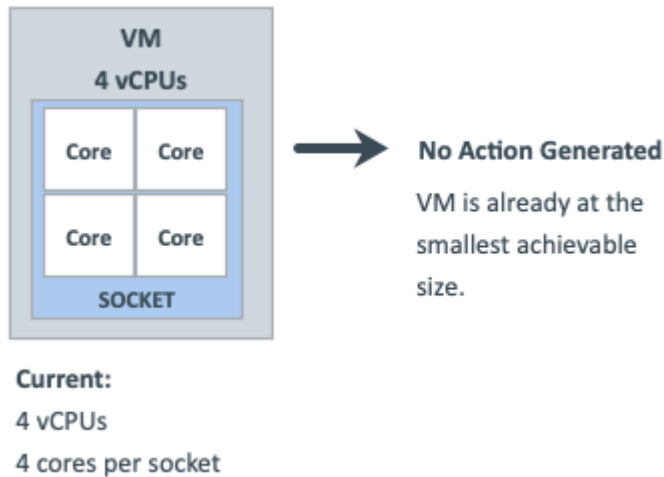
- You require Turbonomic to leave the VM cores per socket configuration unchanged for operational policy reasons (such as compliance with an application support contract policy).
- You have VMs that need to upsize non-disruptively to meet rising application demand.
- You have VMs with even numbers of cores per socket and are required to scale in even increments of vCPUs.

For example, a VM currently has 1 socket and 4 cores per socket, and applies a policy that changes sockets and preserves VM cores per socket. Turbonomic has determined that the VM requires a change in compute capacity of 1 vCPU.

- To increase compute capacity by 1 vCPU, a resize up action adds 1 socket. Because this new socket must have 4 cores to preserve VM cores per socket, the end result is 2 sockets with a total of 8 vCPUs.



- It is not possible to reduce compute capacity by 1 vCPU because the VM is already at the smallest achievable size. Therefore, no action generates.



### Change Sockets and Specify Cores Per Socket

In this scaling option, Turbonomic adds or removes compute resources by changing VM sockets in increments of 1, and reconfigures VM cores per socket according to the value that you specify.

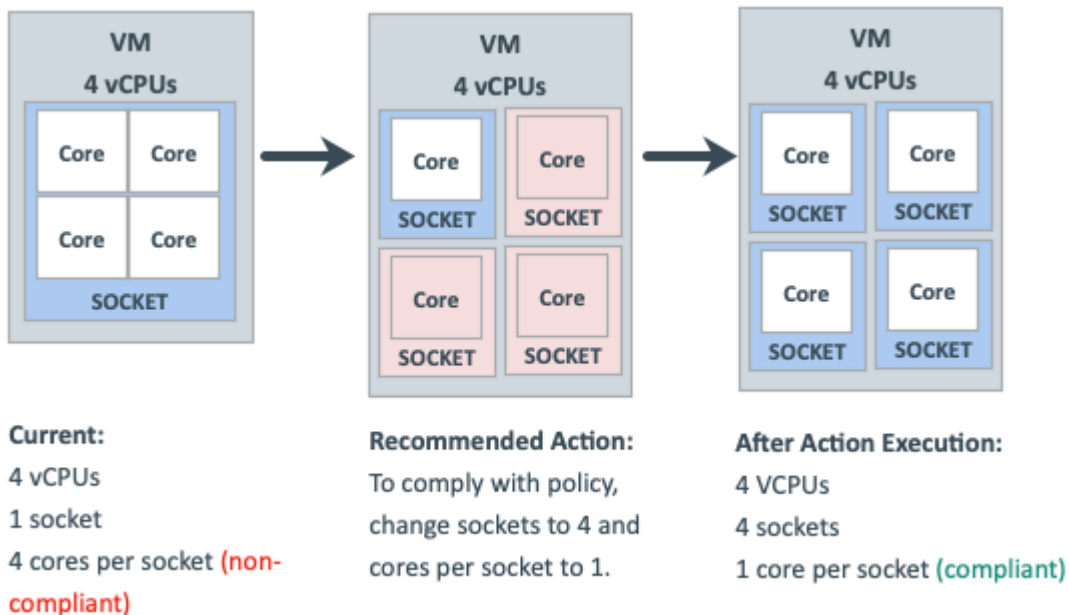
This scaling option is ideal under the following scenarios:

- You require any odd number vCPUs for a VM to be an even number, by setting an even number of cores per socket.

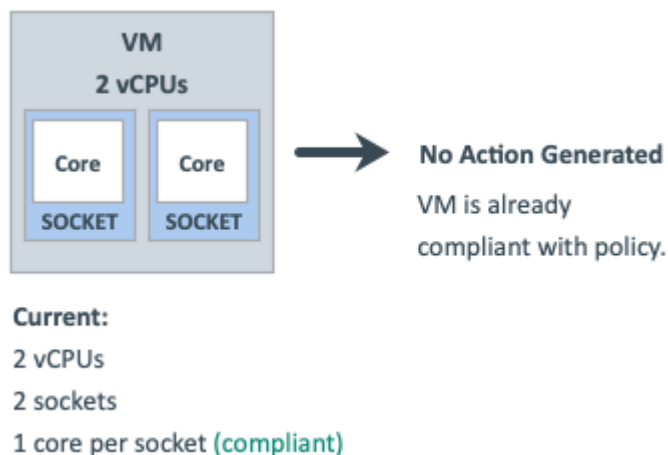
- You want a quick, script-less bulk disruptive conversion of VMs to a specific cores per socket without negatively impacting compute capacity (vCPUs).
- You have older Guest OSes and applications that are sensitive to vCPU architecture changes that could result in power-on issues or kernel panics/BSODs. Some workloads require manual help with such changes so always test certain classes of applications and OSes before enabling any automation that changes the vCPU architecture. Use the Turbonomic knowledge of the application domain and Guest OS to scope them out of policies.

For example, a VM currently has 1 socket and 4 cores per socket, and applies a policy that changes sockets and enforces the user-specified 1 core per socket. Turbonomic has determined that the VM is already optimally sized, so a resize action is not necessary.

- Since the VM is in violation of policy, Turbonomic changes sockets from 1 to 4, and cores per socket from 4 to 1.



- When the VM is compliant with policy, no action generates.



### Scaling Option: Change Cores Per Socket

In this scaling option, Turbonomic adds or removes compute resources by changing the VM cores per socket.



- If a VM requires a change to compute resources, Turbonomic generates a resize vCPU action that considers the current socket value (if the 'Preserve existing VM sockets' option is set), respects the user-specified socket value, or matches VM sockets to the host socket value. If the VM's current socket value violates a policy (i.e., does not match the user-specified or host socket value), Turbonomic reconfigures the VM's socket value as part of action execution, thereby bringing the VM into compliance with the policy while at the same time providing the required change to compute resources.
- If a VM is already optimally sized, but its current socket value violates a policy, Turbonomic generates a reconfigure vCPU action to change the sockets to the user-specified or host socket value, thereby bringing the VM into compliance with the policy.

Older Guest OSes and applications may be sensitive to vCPU architecture changes that could result in power-on issues or kernel panics/BSODs. Some workloads require manual help with such changes so always test certain classes of applications and OSes before enabling any automation that changes the vCPU architecture. Use the Turbonomic knowledge of the application domain and Guest OS to scope them out of policies.

### Change Cores Per Socket and Preserve VM Sockets

In this scaling option, Turbonomic adds or removes compute resources by changing the VM cores per socket in increments of 1, and preserves VM sockets.

This scaling option is ideal under the following scenarios:

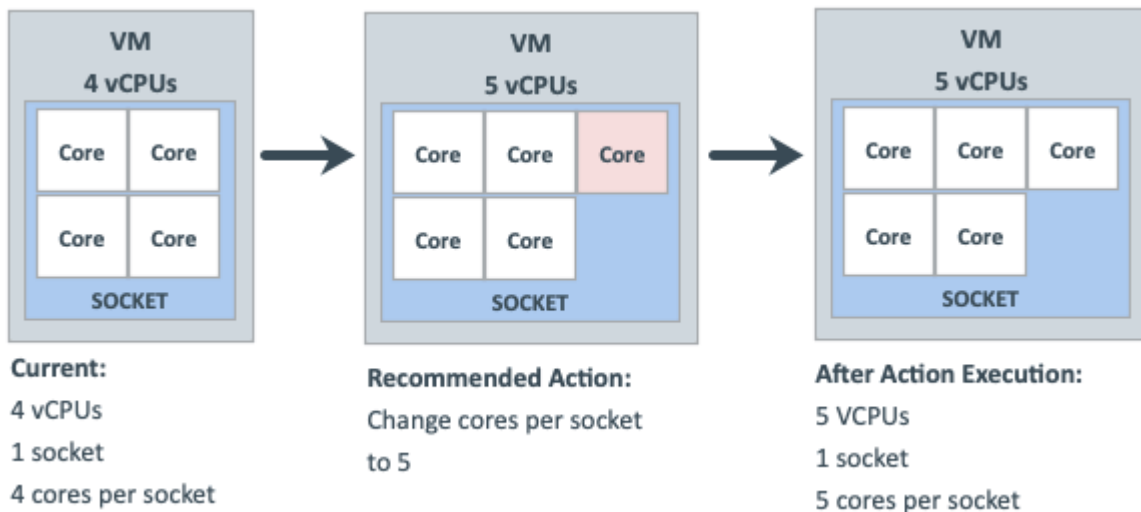
- You require Turbonomic to leave the VM sockets configuration unchanged for operational policy reasons (such as socket-based licensing or compliance with an application support contract policy).
- You have VDI VMs that are at their maximum Guest OS socket limitation, but require more compute resources.
- You have VMs that are configured with NUMA considerations.

#### Note:

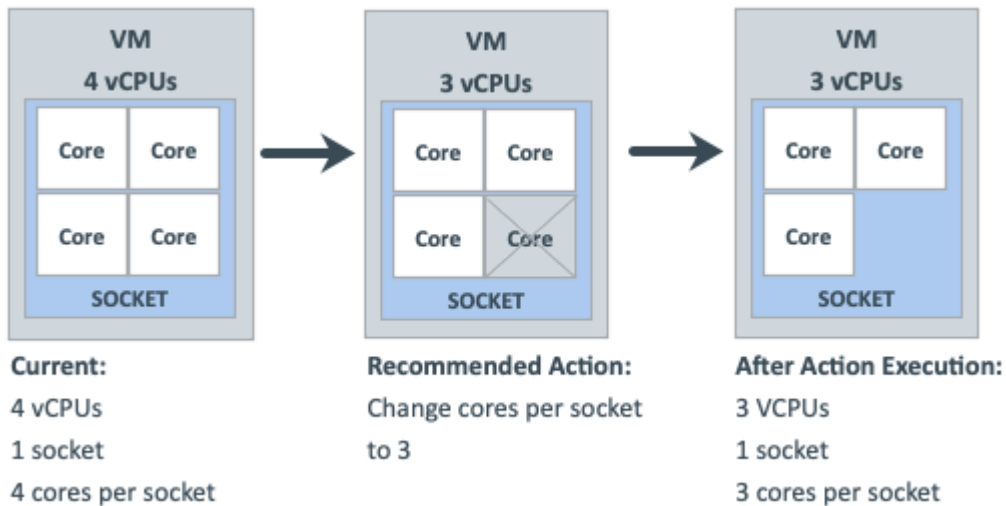
You can also use the 'Match Host Sockets' scaling option (discussed below) for NUMA sensitive VMs.

For example, a VM currently has 1 socket and 4 cores per socket, and applies a policy that changes cores per socket and preserves VM sockets. Turbonomic has determined that the VM requires a change in compute capacity of 1 vCPU.

- To increase compute capacity by 1 vCPU, a resize up action changes cores per socket from 4 to 5.



- To reduce compute capacity by 1 vCPU, a resize down action changes cores per socket from 4 to 3.



### Change Cores Per Socket and Match Host Sockets

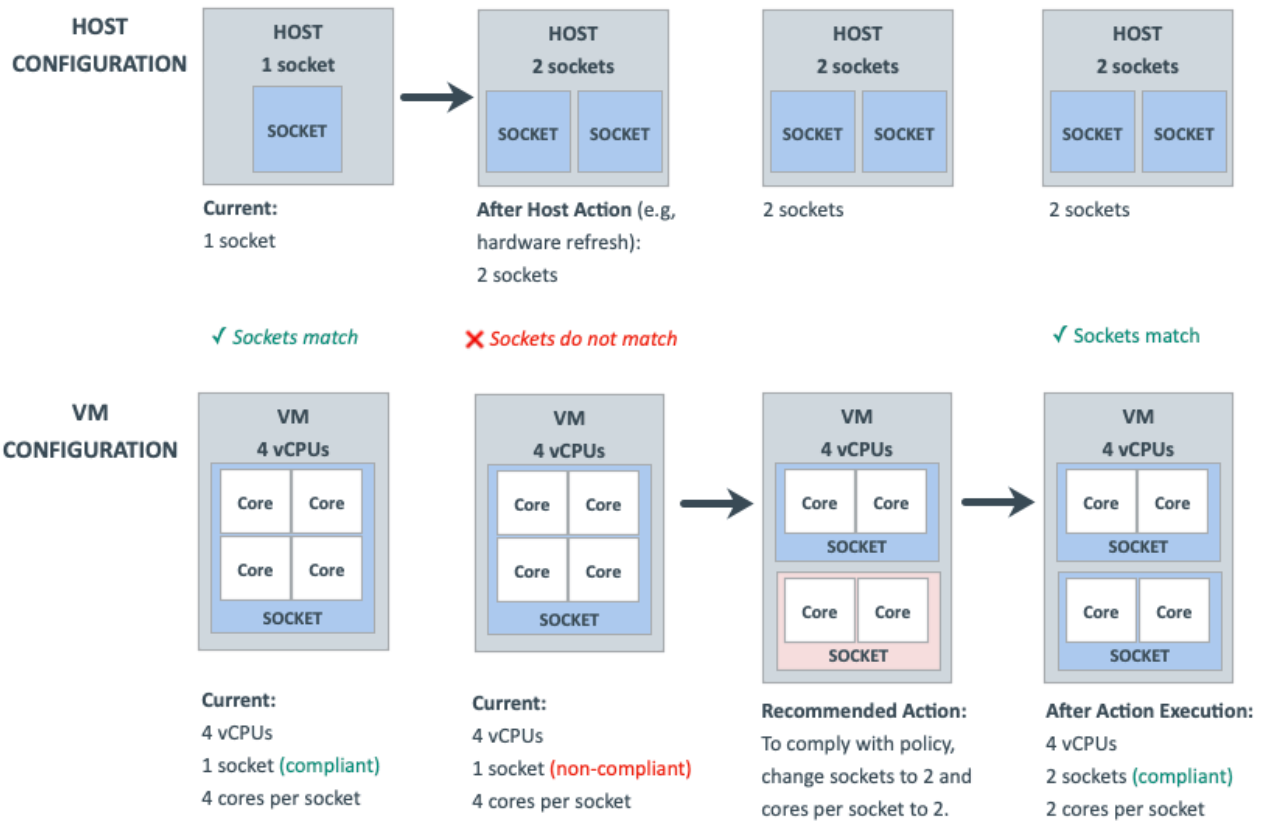
In this scaling option, Turbonomic reconfigures VM sockets to match the number of host sockets, thereby balancing vCPUs evenly across physical sockets. It also changes VM cores per socket to maintain the same compute capacity (vCPU).

This scaling option is ideal under the following scenarios:

- You have large VMs that may realize a performance benefit from reflecting the physical host CPU architecture within the Guest OS so that the application can optimize thread memory access to within a NUMA node.
- You have NUMA sensitive VMs that are migrating between hosts with different CPU architectures. Turbonomic can place the VMs on the best host and then generate an action to reconfigure the VMs to match the host sockets automatically. You can attach a schedule to the policy to automate disruptive reconfigure actions within a maintenance window.

For example, a VM currently has 1 socket and 4 cores per socket, and is on a host with 1 socket. The VM applies a policy that changes cores per socket and matches host sockets. Turbonomic has determined that the VM is already optimally sized, so a resize action is not necessary.

When the host socket value changes from 1 to 2, the VM is suddenly in violation of policy. To bring the VM into compliance while maintaining the same vCPU capacity (since the VM is already optimally sized), Turbonomic must distribute 4 cores between 2 sockets. The end result is 2 sockets and 2 cores per socket.



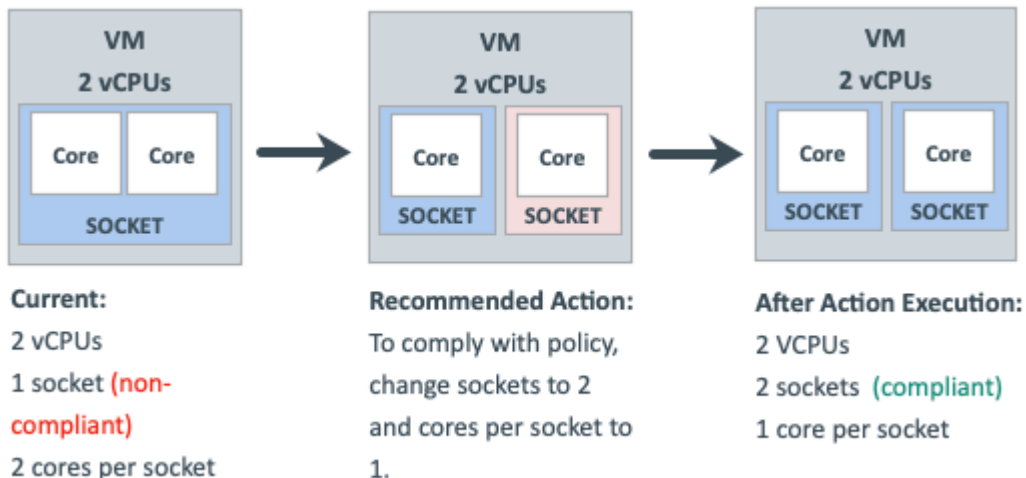
### Change Cores Per Socket and Specify Sockets

In this scaling option, Turbonomic reconfigures VM sockets according to the value that you specify, and changes VM cores per socket to maintain the same compute capacity (vCPU).

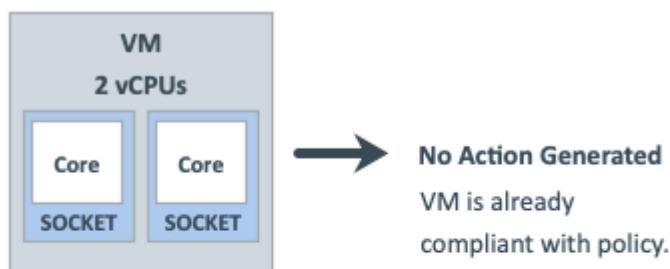
This scaling option is ideal if you have VMs that require a specific socket value for operational policy reasons (such as socket-based licensing or compliance with an application support contract policy).

For example, a VM currently has 1 socket and 2 cores per socket, and applies a policy that changes cores per socket and enforces the user-specified 2 sockets. Turbonomic has determined that the VM is already optimally sized, so a resize action is not necessary.

- Since the VM is in violation of policy, Turbonomic changes sockets from 1 to 2, and cores per socket from 2 to 1.



- When the VM is compliant with policy, no action generates.



**Current:**

2 vCPUs

2 sockets (compliant)

1 core per socket

### Scaling Option: Change MHz Legacy Behavior

In this scaling option, Turbonomic adds or removes compute resources in increments of MHz (1800 MHz by default).

If a VM requires a change to compute resources, Turbonomic generates a resize vCPU action that assumes 1 core per socket, regardless of the VM's actual cores per socket.

If Turbonomic discovers the actual number of cores per socket as part of action execution, it adjusts the action accordingly.

For example, a VM currently has 4 vCPUs with 2 sockets and 2 cores per socket. Turbonomic may generate an action to resize from 4 to 5 vCPUs. However, as part of action execution, the VM socket count changes from 2 to 3, so the end result is 6 vCPUs. Conversely, the same VM may have an action to resize from 4 to 3 vCPUs, but nothing changes as part of action execution.

### Hypervisor Support

For **VMware vSphere**, Turbonomic supports all vCPU scaling options, including changing a VM's number of sockets or cores per socket. Increasing the number of sockets is non-disruptive if CPU hot-add is enabled on a VM, while reducing the socket count always requires a restart and is therefore disruptive.

For **Hyper-V** and **Nutanix AHV**, cores per socket and hot-add features have varying degrees of support.

| vCPU Scaling Option                                    | vSphere   | Hyper-V                     | Nutanix AHV (Single Core)   | Nutanix AHV (Multi Core)    |
|--------------------------------------------------------|-----------|-----------------------------|-----------------------------|-----------------------------|
| Change virtual CPUs                                    | Supported | Supported                   | Supported                   | Not supported by hypervisor |
| Change sockets – Preserve existing VM cores per socket | Supported | Supported                   | Supported                   | Supported                   |
| Change sockets – User specified cores per socket       | Supported | Not supported by hypervisor | Not supported by hypervisor | Not supported by hypervisor |

| <b>vCPU Scaling Option</b>                             | <b>vSphere</b> | <b>Hyper-V</b>                                                                   | <b>Nutanix AHV (Single Core)</b> | <b>Nutanix AHV (Multi Core)</b> |
|--------------------------------------------------------|----------------|----------------------------------------------------------------------------------|----------------------------------|---------------------------------|
| Change cores per socket – Preserve existing VM sockets | Supported      | Not supported by hypervisor                                                      | Not supported by Turbonomic      | Not supported by Turbonomic     |
| Change cores per socket – Match host sockets           |                | <b>Note:</b><br>Turbonomic assumes one core per socket and only changes sockets. |                                  |                                 |
| Change cores per socket – User specified sockets       |                |                                                                                  |                                  |                                 |

## Tie Breakers

When a single VM applies multiple conflicting policies, Turbonomic uses the following tie breakers that follow the principle of least disruptive and most conservative:

- vCPU Scaling Control

"Sockets" wins over "Cores per socket" wins over "Virtual CPU" wins over "MHz legacy behavior".

### **Note:**

Policies created before the introduction of vCPU scaling controls (i.e., any policy before version 8.5.7) will continue to use the "MHz legacy behavior" option but will not be enforced when policy conflicts arise. You can remove these policies or update them to use the newer scaling controls.

- Sockets setting

"Preserve existing VM cores per socket" wins over "User-specified core per socket".

- Cores Per Socket setting

"Preserve existing VM sockets" wins over "User-specified socket" wins over "Match host sockets".

- User-specified value

The lowest value wins.

- Increment Size value

The lowest value wins.

For example, assume a VM belonging to two groups that apply different policies. Policy A changes cores per socket and matches host sockets, while Policy B changes sockets and preserves cores per socket. In this scenario, the VM applies Policy B. Changing sockets wins over changing cores per socket because it is less disruptive.

To see which policies are in effect after the tie-break decision, set the scope to a VM or group of VMs and then click the Policies tab.

## Policy Cookbook

### **Tips:**

- Use the following filters when searching for or creating VM groups:
  - Number of vCPUs
  - Number of Sockets
  - Cores per Socket

- Target Type
- Hot-Add Enabled
- For the least disruptive on-demand upsize of vCPU, enable hot-add on the VM and change sockets while preserving cores per socket.
- For the most precise compute resource management, change cores per socket.
- For NUMA considerations, change cores per socket and match host sockets.
- Check Guest OS application and license compatibility when changing vCPU architecture and before automating actions.

#### How to...

- Manage VM compute capacity by changing the number of vCPUs in increments of 2.

A VM will be reconfigured if required to use 1 core per socket, and resized by changing sockets. Actions are disruptive if the VM does not already have 1 core per socket or if hot-add is not enabled.

1. Create a group of VMs that can have 1 core per socket and scale in sockets.
2. Assign the group a policy with the following settings:

- vCPU Scaling Controls
  - Change: Virtual CPU
  - Increment size: 2
- (Optional) vCPU Resize Min/Max Threshold

- Reconfigure all odd-numbered vCPU VMs to be even-numbered, and then manage compute in even numbers of CPUs.

A VM will be reconfigured if required to use 2 cores per socket, and resized by changing sockets. Actions are disruptive if the VM does not already have 2 cores per socket or if hot-add is not enabled.

1. Create a group of VMs that can have 2 cores per socket and scale in sockets.
2. Assign the group a policy with the following settings:

- vCPU Scaling Controls
  - Change: Sockets
  - User specified cores per socket: 2
- (Optional) vCPU Resize Min/Max Threshold

- Ensure that large VMs always balance their vCPU cores across all physical host sockets (for example, NUMA VMs and Database Server VMs).

A VM will be reconfigured if its socket count does not match the host socket count. The cores per socket count may be adjusted to maintain the overall compute capacity (number of vCPUs). Resize actions are disruptive because cores per socket will change. Reconfigure actions are non-disruptive if VM sockets are increasing, hot-add is enabled, and there are no changes to cores per socket.

1. Create a group of VMs using the filters that you require to identify typically larger VMs.
2. Assign the group a policy with the following settings:

- vCPU Scaling Controls
  - Change: Cores per socket
  - Sockets: Match host sockets
- (Optional) vCPU Resize Min/Max Threshold

- Keep VMs to 2 sockets only and manage compute by changing cores.

VMs in the group will be reconfigured to 2 sockets if required, and resized by changing the cores per socket count while keeping the sockets fixed at 2, thus ensuring compliance with socket-based licensing. Resize actions are disruptive because cores per socket will change. Reconfigure actions are

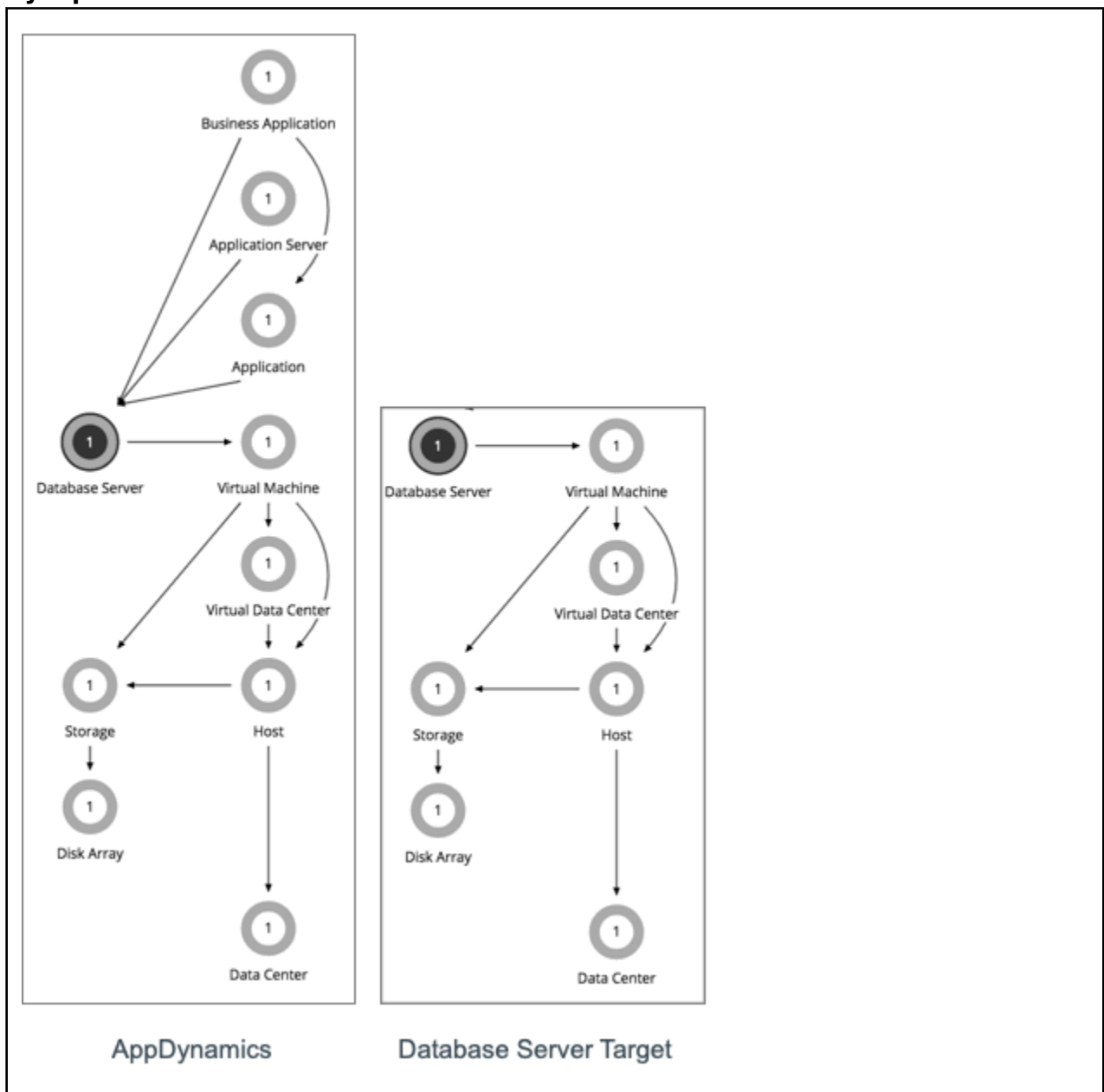
non-disruptive if VM sockets are increasing, hot-add is enabled, and there are no changes to cores per socket.

1. Create a VM group containing the socket-licensed VMs.
2. Assign the group a policy with the following settings:
  - vCPU Scaling Controls
    - Change: Cores per socket
    - User specified sockets: 2
  - (Optional) vCPU Resize Min/Max Threshold

## Database Server (On-prem)

For on-prem, a Database Server is a database discovered through one of the associated database application targets or through APM solutions.

### Synopsis



| <b>Synopsis</b>     |                                                                                                                                                                                                                                                           |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | On-prem Database Servers have unlimited budget.                                                                                                                                                                                                           |
| Provides:           | <ul style="list-style-type: none"> <li>• Response Time, Transactions, DBmem, Cache Hit Rate, and TransactionLog to end users</li> <li>• Connections to Application Components</li> </ul>                                                                  |
| Consumes:           | VM resources, including VCPU, VMem, and VStorage                                                                                                                                                                                                          |
| Discovered through: | <ul style="list-style-type: none"> <li>• AppDynamics targets</li> <li>• Database Server targets</li> <li>• Dynatrace MySQL and SQL Server processes</li> <li>• NewRelic Infrastructure Integration (NRI): MySQL, SQL Server, MongoDB, OracleDB</li> </ul> |

**Monitored Resources**

Turbonomic monitors the following resources for an on-prem Database Server:

- Virtual Memory  
Virtual Memory is the measurement of memory utilized by the entity.
- Transactions  
Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.
- Database Memory  
Database memory (or DBMem) is the measurement of memory utilized by a Database Server.
- Connections  
Connection is the measurement of Database Server connections utilized by applications.
- DB Cache Hit Rate  
DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

**Actions**

**Resize**

Resize the following resources:

- Connections  
Turbonomic uses connection data to generate memory resize actions for on-prem Database Servers.
- Database memory (DBMem)  
Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.  
A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.



When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.

- Transaction log

Resize actions based on the transaction log resource depend on support for virtual storage in the underlying hypervisor technology.

Currently, Turbonomic does not support resize actions for Oracle and Database Servers on the Hyper-V platform (due to the lack of API support for virtual storage).

## On-prem Database Server Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

| Action                 | Default Mode |
|------------------------|--------------|
| Resize                 | Manual       |
| Resize DBMem (Up/Down) | Manual       |

You can use [Action Scripts](#) and third-party orchestrators (such as ServiceNow) for action orchestration.

### Transaction SLO

Transaction SLO determines the upper limit for acceptable transactions per second. When the number of transactions reaches the given value, Turbonomic sets the risk index to 100%.

| Attribute              | Default Setting/Value                                                                                         |
|------------------------|---------------------------------------------------------------------------------------------------------------|
| Enable Transaction SLO | Off                                                                                                           |
| Transaction SLO        | None<br>If you enable SLO, Turbonomic uses the default value of 10. You can change this to a different value. |

### Response Time SLO

Response time SLO determines the upper limit for acceptable response time (in milliseconds). If response time reaches the given value, Turbonomic sets the risk index to 100%.

| Attribute                | Default Setting/Value                                                                                           |
|--------------------------|-----------------------------------------------------------------------------------------------------------------|
| Enable Response Time SLO | Off<br>Turbonomic estimates SLO based on monitored values.                                                      |
| Response Time SLO [ms]   | None<br>If you enable SLO, Turbonomic uses the default value of 2000. You can change this to a different value. |

## DBMem Utilization

The utilization that you set here specifies the percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute             | Default Value |
|-----------------------|---------------|
| DBMem Utilization (%) | 100           |

For example, a value of 80 means that Turbonomic considers 80% utilization to be 100% of capacity. Turbonomic recommends actions that avoid utilization beyond the given value.

## DBMem Scaling Increment

This increment specifies how many units to add or subtract when scaling DBMem.

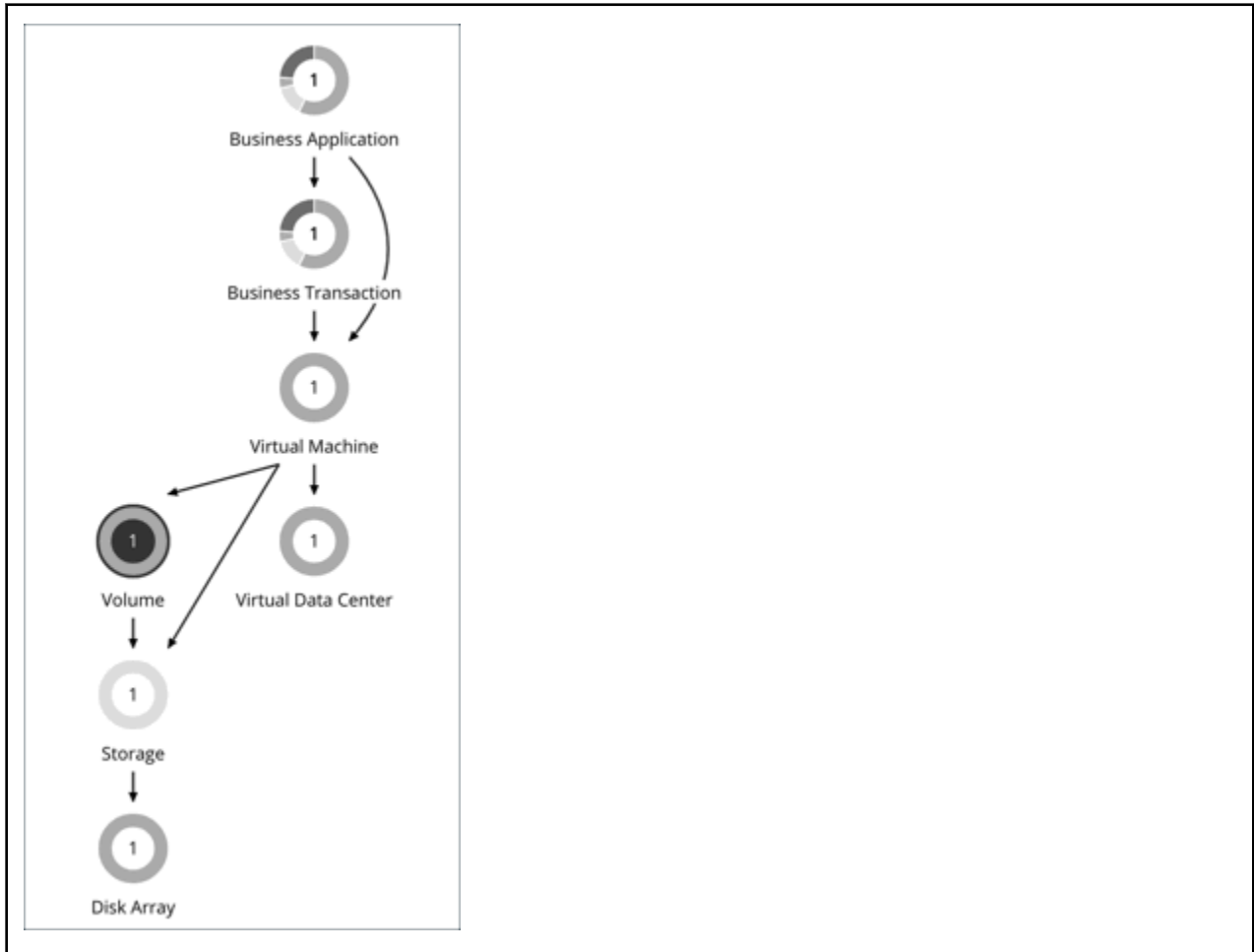
| Attribute                    | Default Value |
|------------------------------|---------------|
| DBMem Scaling Increment (MB) | 128           |

Do not set the increment value to be lower than what is necessary for the database server to operate. If the increment is too low, then it's possible there would be insufficient DBMem. When reducing allocation, Turbonomic will not leave a database server with less than the increment value. For example, if you use the default 128, then Turbonomic cannot reduce DBMem to less than 128 MB.

## Volume (On-prem)

On-prem volumes represent VM disks discovered by hypervisor targets. A VM will have one volume for each configured disk and another volume (representing the configuration) that always moves with Disk 1.

## Synopsis



## Synopsis

|                     |                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | An on-prem volume gains its budget by selling resources to the VMs that it serves.                                                                                                                                                                    |
| Provides:           | Storage resources for VMs to use.<br>Set the scope to a volume and view the Entity Information chart to see a list of VM-related files (such as VMDKs) contained in the volume.<br>Set the scope to a VM to see a list of volumes attached to the VM. |
| Consumes:           | Datacenter resources                                                                                                                                                                                                                                  |
| Discovered through: | Hypervisor targets                                                                                                                                                                                                                                    |

## Actions

### • Move

Move a VM's volume (virtual storage) due to excess utilization of the current datastore, or for more efficient utilization of datastores in the environment.

Points to consider:

- The default global policy includes a setting that directs Turbonomic to use relevant metrics when analyzing and recommending actions for volumes. For details, see [Enable Analysis of On-prem Volumes](#).
- Turbonomic will not recommend moving a volume to a datastore that is currently in maintenance mode. Any volume in that datastore should move to an active datastore (for example, via vMotion).

- **Reconfigure**

Reconfigure a VM's volume (virtual storage) to comply with placement policies.

## On-prem Volume Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Placement Policies

By default, all on-prem volumes associated with a storage will move together rather than independently. You can create placement policies to place individual volumes on groups of storage. To ensure successful placement, be sure to also turn on the setting `Enable Analysis of On-prem Volumes` in the default global policy.

For more information, see [Creating Placement Policies](#) and [Enable Analysis of On-prem Volumes](#)

### Action Automation and Orchestration

| Action | Default Mode |
|--------|--------------|
| Move   | Manual       |

### Cloud Storage Tiers

This policy setting works with plans that simulate migration of on-prem volumes to the cloud. When you create the policy, be sure to set the scope to on-prem volumes and then select the cloud storage tiers that they can migrate to. Turbonomic treats these tiers as constraints when you run a Migrate to Cloud plan that includes the volumes defined in the policy.

| Attribute           | Default Value |
|---------------------|---------------|
| Cloud Storage Tiers | None          |

Click **Edit** to set your preferences. In the new page that displays, expand a **cloud tier** (a family of instance types, such as *Premium*) to see individual instance types.

Select your preferred instance types or cloud tiers, or clear the ones that you want to avoid. After you save your changes, the main page refreshes to reflect your selections.

## Virtual Data Center (Private Cloud)

A virtual data center (vDC) is a collection or pool of resources that groups the resources around specific requirements or business needs. In private cloud environments, Turbonomic discovers the infrastructure that provides resources to the cloud, and the workloads that run on the cloud. To manage these resources, private clouds organize the infrastructure into Provider and Consumer Virtual Data centers.

**Note:**

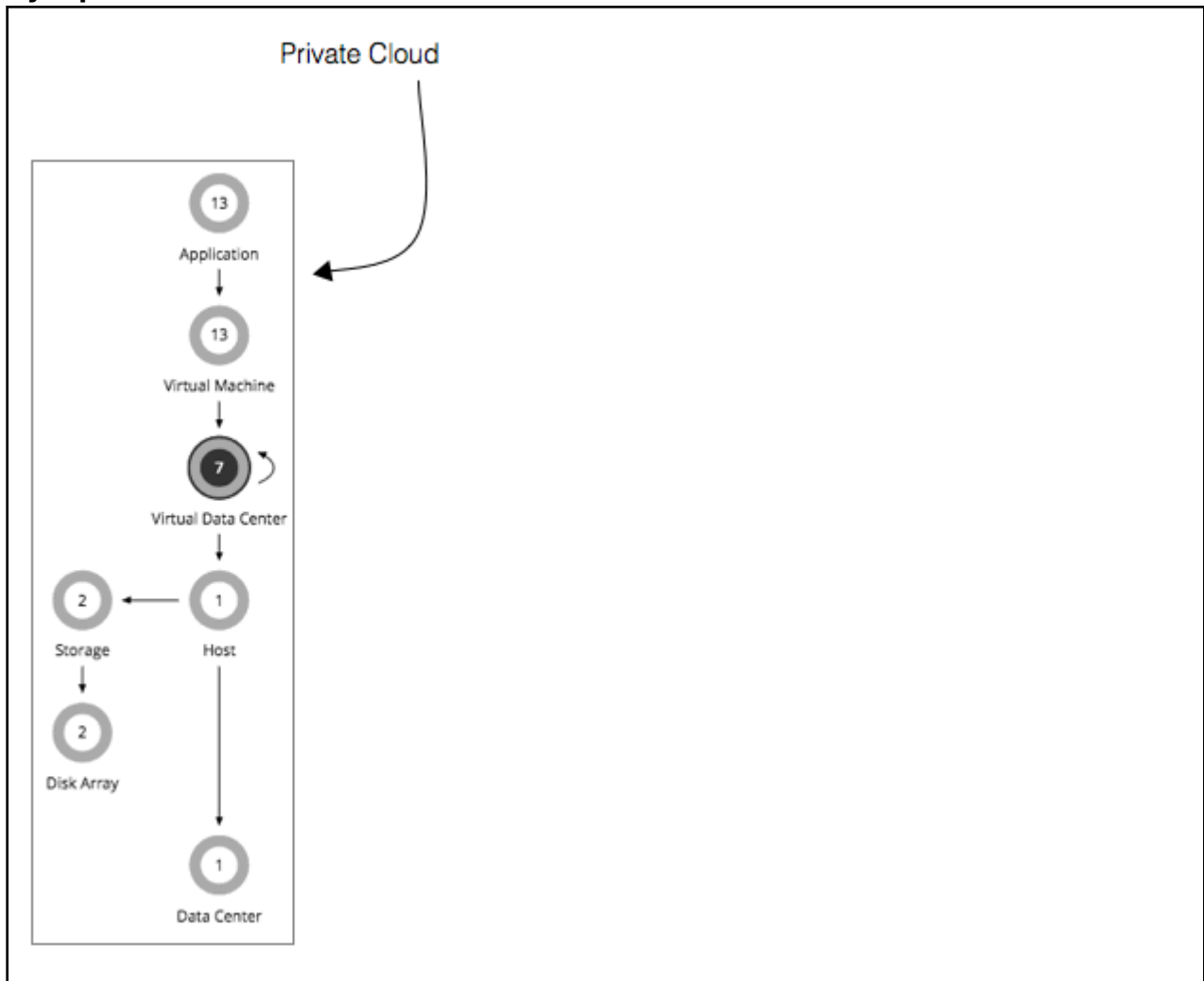
Different targets use different names to refer to Virtual Data centers. In the Turbonomic supply chain, these entities are all represented by Consumer and Provider VDCs, as follows:

| Turbonomic   | vCenter Server        | VMM                   |
|--------------|-----------------------|-----------------------|
| Consumer VDC | Resource Pool (Child) | Tenant or TenantQuota |
| Provider VDC | Resource Pool (Root)  | Cloud                 |

## Provider Virtual Data Centers

A provider virtual data center (vDC) is a collection of physical resources (hosts and data stores) within a cloud stack. The cloud administrator has access to these resources, and defines the data center members. A Provider vDC is created to manage resources that will be allocated to external customers through one or more Consumer vDCs.

### Synopsis



### Synopsis

|           |                                                                                                                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:   | A Provider vDC gains its budget by selling resources to the Consumer vDCs that it hosts. If utilization falls off, the data center loses budget. Ultimately, if the budget isn't enough to pay for the services it consumes, Turbonomic will recommend decommissioning the Provider vDC. |
| Provides: | Physical resources such as hosts and datastores to Consumer vDCs.                                                                                                                                                                                                                        |

## Synopsis

|                     |                                                                 |
|---------------------|-----------------------------------------------------------------|
| Consumes:           | Hosts and datastores from the physical infrastructure           |
| Discovered through: | Turbonomic discovers vDCs through private cloud stack managers. |

## Monitored Resources

Turbonomic monitors the following resources for a Provider vDC:

- Memory (Mem)  
The utilization of the Data center's memory reserved or in use.
- CPU  
The utilization of the Data center's CPU reserved or in use.
- Storage  
The utilization of the storage attached to the Provider vDC.

## Actions

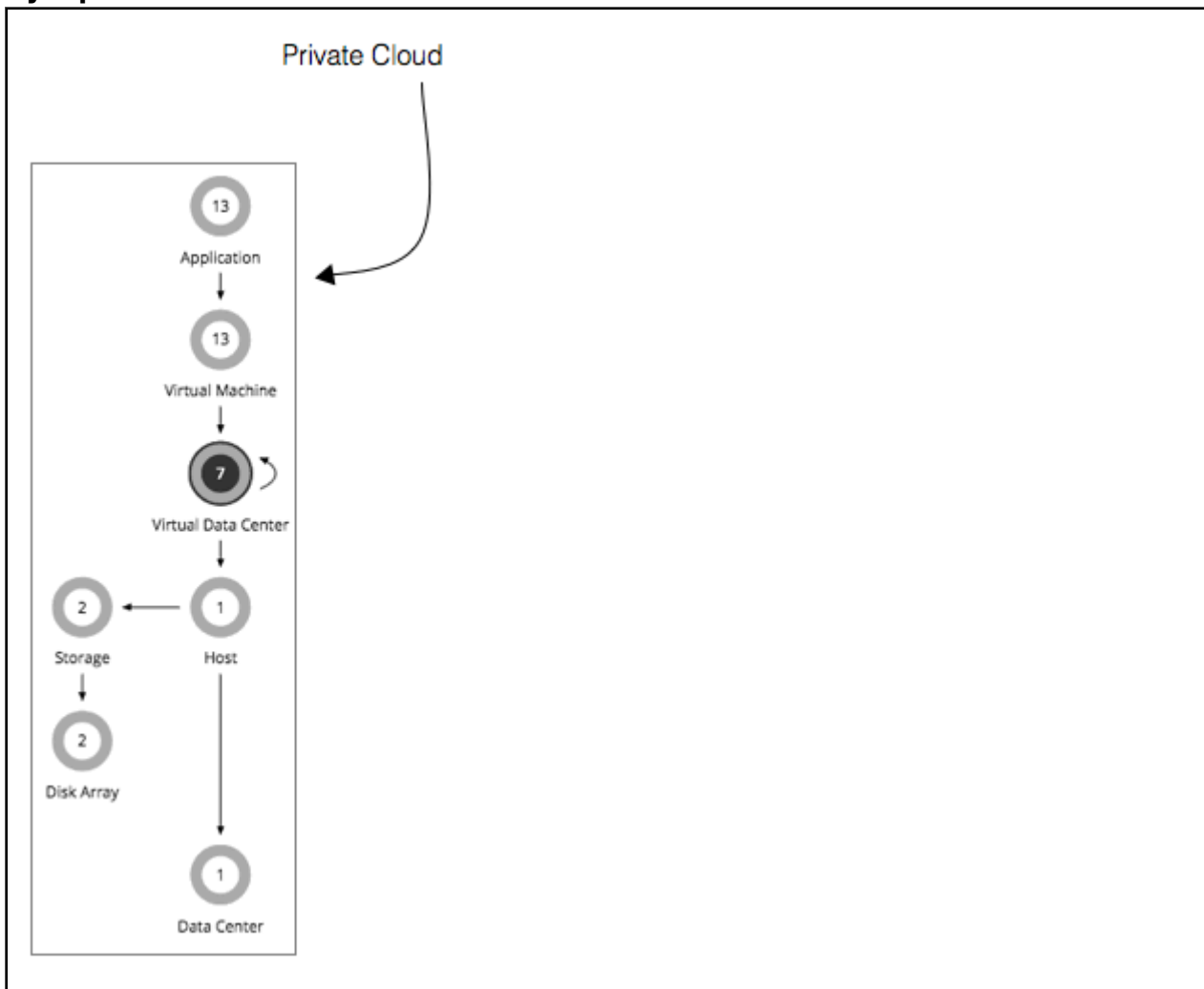
### None

Turbonomic does not recommend actions for a Virtual Data Center. Instead, it recommends actions for the entities that provide resources to the Virtual Data Center.

## Consumer Virtual Data Centers

A Consumer Virtual Data center (vDC) is a collection of resources that are available for external customers to manage workload through the private cloud. It is an environment customers can use to store, deploy, and operate virtual systems. Consumer Data centers use the resources supplied by a Provider Data center.

## Synopsis



## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | <p>A Consumer vDC gains its budget as a function of its activity. The higher the utilization of the vDC, the more Turbonomic assumes the vDC is selling its services to a user.</p> <p>If utilization is high enough on a Consumer vDC, Turbonomic can increase resources for the vDC. If utilization falls off, Turbonomic can reduce resource capacity, or ultimately recommend terminating the vDC.</p> <p>Turbonomic can also resize VMs through the Consumer vDC in response to changes in VM utilization.</p> |
| Provides:           | Resources to host virtual systems.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Consumes:           | Provider vDC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Discovered through: | Turbonomic discovers vDCs through cloud stack managers.                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

While users can see some of the physical resources that support the Consumer vDC, consumer-level users cannot modify these physical resources. Users of Consumer vDCs make changes to how the virtual devices are deployed in that environment, but they must ask the Provider vDC administrator to add more

physical resources to be used by the Consumer vDC. Likewise, Turbonomic can change resources on the VMs running in the vDC, but it does not make any changes to physical resources through this vDC.

## Monitored Resources

Turbonomic monitors the following resources for a Consumer vDC:

- Memory (Mem)  
The utilization of the Data center's memory reserved or in use.
- CPU  
The utilization of the Data center's CPU reserved or in use.
- Storage  
The utilization of the storage attached to the Consumer vDC.

## Actions

Turbonomic does not recommend actions to perform on a Consumer vDC. Instead, it recommends actions to perform on the entities running in the Provider vDC.

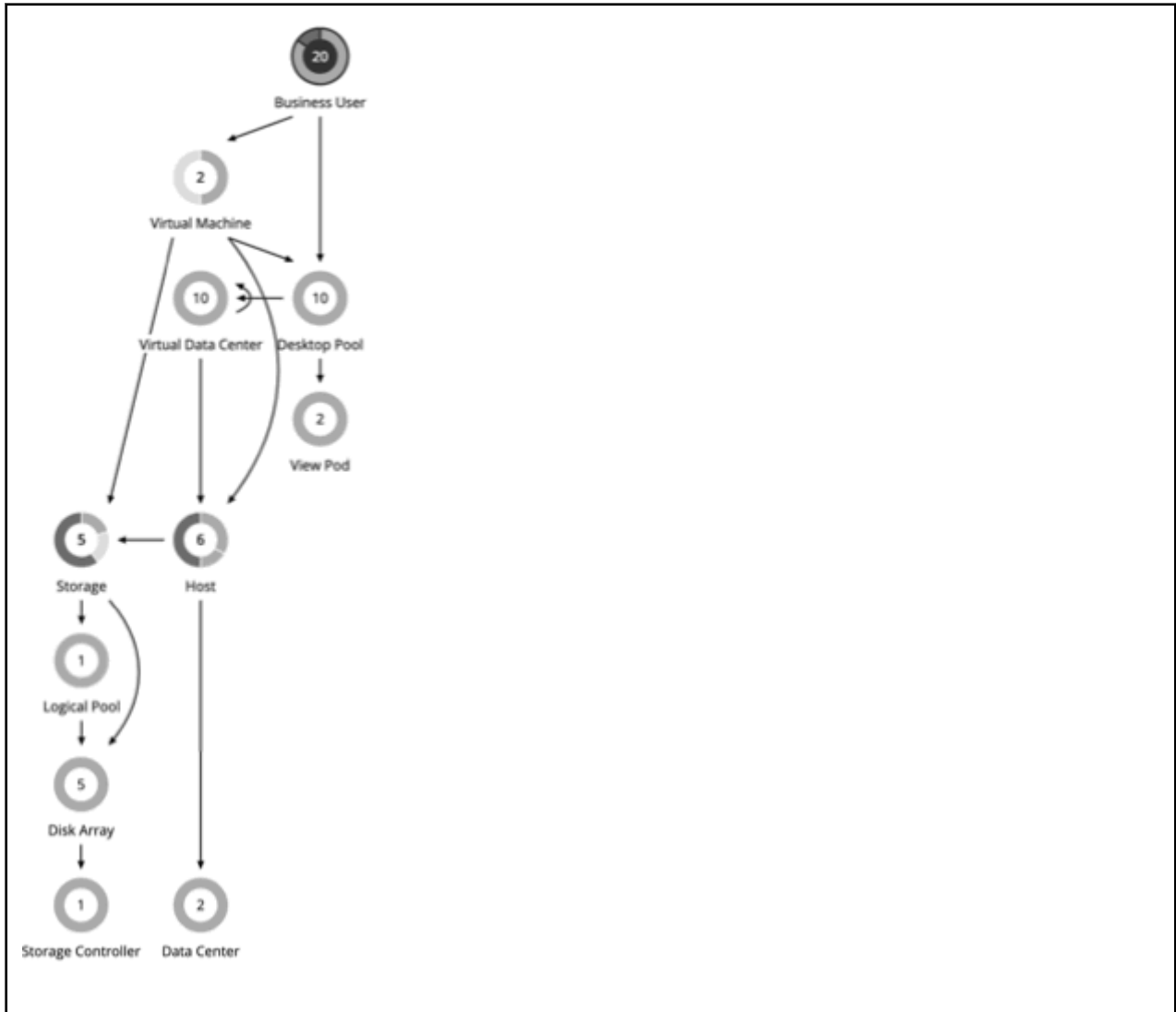
## Business User

For Virtual Desktop Infrastructure (VDI) environments, a Business User is a user account that is entitled to launch one or more active VDI sessions. As it discovers desktop pools, Turbonomic creates Business User entities for each user that is entitled to a pool. One business user can be entitled to more than one desktop pool.

To properly work with Business User entities, Turbonomic discovers user information through the LDAP server that manages users for the VDI environment. Note that the account Turbonomic uses to connect to the LDAP server must be trusted for the same domains as are the users in your environment.



## Synopsis



The Supply Chain shows relationships of Business Users to Desktop Pools and also to VMs. One Business User can have access to multiple Desktop Pools. When a Business User has an active session, the Supply Chain shows a direct link between the user and the VM that hosts the session. However, Turbonomic does not consider this direct connection when analyzing compute resources. Instead, Business Users utilize Desktop Pool resources, and the Desktop Pools use compute resources from the underlying Virtual Datacenters.

### Synopsis

|           |                                       |
|-----------|---------------------------------------|
| Budget:   | A Business User has unlimited budget. |
| Provides: | N/A                                   |

## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Consumes:           | <p>Resources from the underlying desktop pools:</p> <ul style="list-style-type: none"><li>• Sessions</li><li>• Pool Memory</li><li>• Pool Storage</li><li>• Pool CPU</li></ul> <p>When a Business User has an active session, the Supply Chain shows it in relation to the VM that hosts the session. The Business User consumes the VM's compute resources to support the session requirements for ImageCPU, ImageMem, and ImageStore resources.</p> |
| Discovered through: | <p>The LDAP server that manages these users. You can specify the LDAP server as part of the target configuration, or Turbonomic can discover it in association with the VDI target.</p>                                                                                                                                                                                                                                                               |

## Monitored Resources

Turbonomic monitors the following resources for a Business User:

- ImageCPU  
CPU utilization, as a percentage of CPU capacity for the user's desktop image or images.
- ImageMem  
Memory utilization, as a percentage of Memory capacity for the user's desktop image or images.
- ImageStorage  
Storage utilization, as a percentage of storage capacity for the user's desktop image or images.

## Business User Actions

### Move

Move a Business User between desktop pools to address:

- Resource congestion on the image  
When utilization is consistently near capacity for image resources, Turbonomic can recommend moving a Business User to a desktop pool that serves larger images.
- Resource congestion on the desktop pool  
When utilization is consistently near capacity for the desktop pool, Turbonomic can recommend moving a Business User to a desktop pool that has more available resources.

### Note:

To support moves, you must configure placement policies that merge *similarly configured* desktop pools. For details, see [Desktop Pool Placement Policies](#).

## Business User Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about Business User actions, see [Business User Actions](#).

| Action | Default Mode |
|--------|--------------|
| Move   | Recommend    |

## Image Target Utilization

Turbonomic tracks utilization of desktop image resources for the Business Users in your Virtual Desktop Infrastructure (VDI) environment.

| Attribute                        | Default Value                                                     |
|----------------------------------|-------------------------------------------------------------------|
| Image CPU Target Utilization     | 70<br>The target utilization as a percentage of CPU capacity.     |
| Image MEM Target Utilization     | 90<br>The target utilization as a percentage of memory capacity.  |
| Image Storage Target Utilization | 90<br>The target utilization as a percentage of storage capacity. |

## Aggressiveness and Observation Period

Turbonomic uses these settings to calculate utilization percentiles. It then recommends actions to improve utilization based on the observed values for a given time period.

### • Aggressiveness

| Attribute      | Default Value   |
|----------------|-----------------|
| Aggressiveness | 95th Percentile |

When evaluating utilization of compute and storage resources, Turbonomic considers a given utilization percentile. For example, assume a 95th percentile. The maximum utilization would be the highest value that 95% of the observed samples fall below.

Using a percentile, Turbonomic can recommend more relevant actions, so that analysis can better exploit elasticity in your environment. A percentile evaluates the sustained resource utilization, and ignores bursts that occurred for a small portion of the samples. You can think of this as aggressiveness of resizing, as follows:

- 100th Percentile – The least aggressive, recommended for critical workloads that need maximum guaranteed performance at all times.
- 95th Percentile (Default) – The recommended setting to achieve maximum performance and savings.
- 90th Percentile – The most aggressive, recommended for non-production workloads that can stand higher resource utilization.

### • Max Observation Period

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 7 Days   |

To refine the calculation of resource utilization, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. (If the database has fewer days' data then it uses all of the stored historical data.)

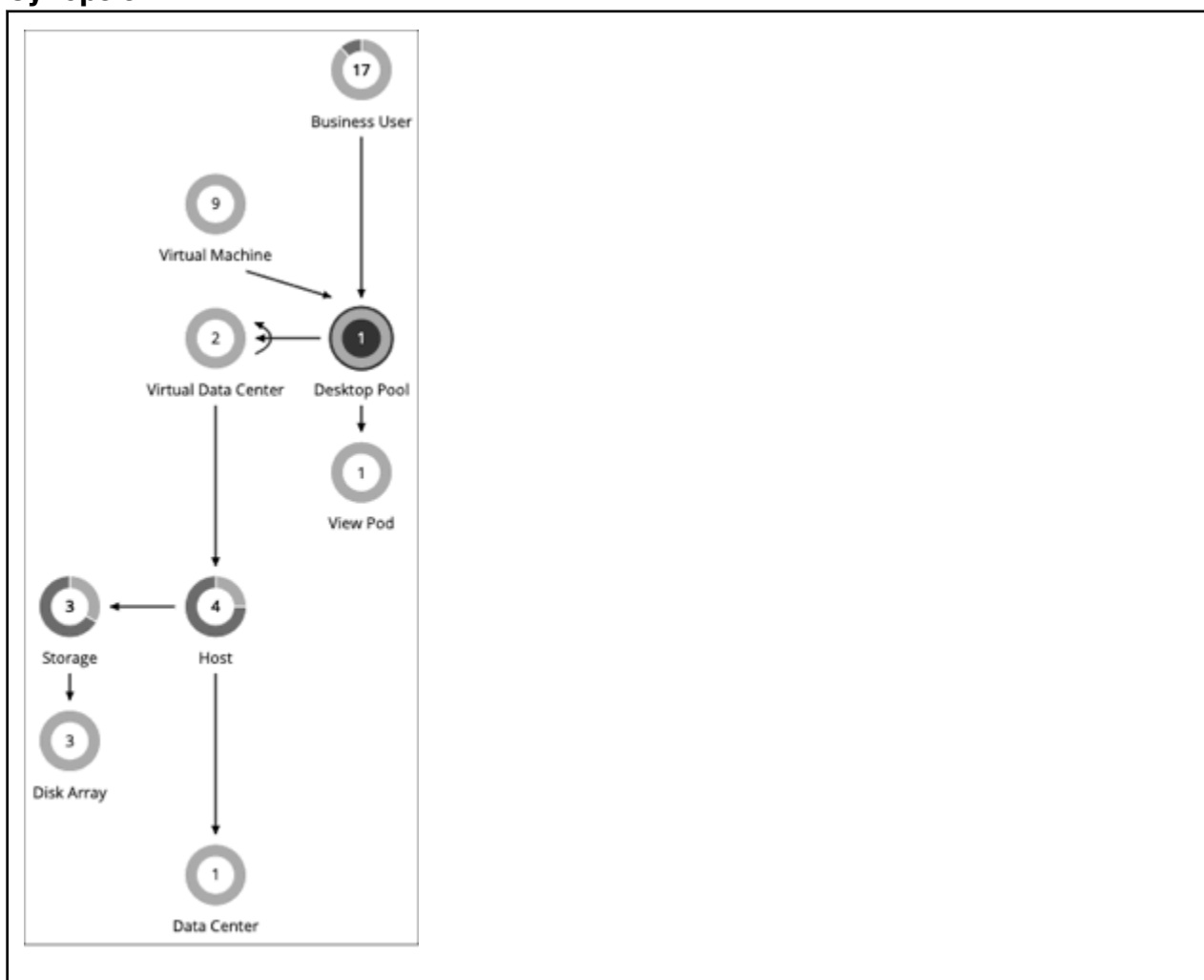
A shorter period means there are fewer data points to account for when Turbonomic calculates utilization percentiles. This results in more dynamic, elastic moves to different Desktop Pools, while a longer period results in more stable or less elastic moves. You can make the following settings:

- Less Elastic – Last 90 Days
- More Elastic – Last 30 Days
- (Default) Most Elastic – Last 7 Days

## Desktop Pool

For Virtual Desktop Infrastructure (VDI) environments, a desktop pool is a collection of desktops that users can select from. The desktop pool can provide logical grouping of desktops according to user roles, assignment type (dedicated or floating), and the source of resources (physical host or VM).

### Synopsis



The desktop pool gets compute and storage resources from the underlying Virtual Datacenter. For VMware Horizon View, the VDI architecture includes one or more vCenter Server instances. When it discovers the Horizon View target, Turbonomic also discovers the supporting vCenter Server instances, and their corresponding Virtual Datacenters. These are the source of compute and storage resources for the associated desktop pools.

| <b>Synopsis</b>     |                                                                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | A Desktop Pool gets its budget by selling resources to Business Users.                                                                                                     |
| Provides:           | Resources for Business Users to use: <ul style="list-style-type: none"> <li>• PoolMEM</li> <li>• PoolCPU</li> <li>• Sessions</li> </ul>                                    |
| Consumes:           | <ul style="list-style-type: none"> <li>• Compute and storage resources from the associated Virtual Datacenters</li> <li>• Sessions from the underlying View Pod</li> </ul> |
| Discovered through: | <p>The VDI management target.</p> <p>For VMware Horizon View, the target is the View Connection Server.</p>                                                                |

**Monitored Resources**

Turbonomic monitors the following resources for a Desktop Pool:

- Pool CPU

The CPU available to the pool that is in use by active sessions.
- Pool Memory

The memory available to the pool that is in use by active sessions.
- Pool Storage

The storage capacity available to the pool that is in use by active sessions.
- Active Sessions

How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.
- Total Sessions

How many active and disconnected (non-terminated) sessions are on the pool, as a percentage of the pool's capacity.

**Actions**

**None**

Turbonomic does not recommend actions for a desktop pool. It recommends actions for the Business Users running active sessions in the pool.

**Desktop Pool Policies**

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

**Action Automation and Orchestration**

**None**

Turbonomic does not recommend actions for a desktop pool. It recommends actions for the Business Users running active sessions in the pool.

## Observation Settings

Turbonomic uses these settings to decide whether to move Business Users from one desktop pool to another.

### • Daily Observation Windows

| Attribute                 | Default Value     |
|---------------------------|-------------------|
| Daily observation windows | 3 windows per day |

When evaluating utilization of pool resources, Turbonomic divides each day into different observation windows, calculates an average for each, and uses the highest value. In this way, Turbonomic can account for high-use periods in the day to base calculations off of the most representative usage of the desktop images.

Assume three observation windows:

| Window | Time range    | Average utilization |
|--------|---------------|---------------------|
| W1     | 00:00 – 08:00 | 10%                 |
| W2     | 08:00 – 16:00 | 80%                 |
| W3     | 16:00 – 24:00 | 40%                 |

Average utilization for this day *without* the benefit of observation windows would be 44%. By using observation windows we can see that the representative utilization of pool resources is closer to 80%. That is because Turbonomic discovers an average utilization of 80% during the high-usage time of day.

When calculating whether to move business users from one desktop pool to another, Turbonomic averages the observation windows over the time you set for the Max Observation Period. For this reason, you should try to set up observation windows that capture the best representation of work habits amongst your business users.

### • Max Observation Period

| Attribute              | Default Value |
|------------------------|---------------|
| Max Observation Period | Last 7 Days   |

To refine the calculation of resource utilization, you can set the sample time to consider. Turbonomic uses historical data from up to the number of days that you specify as a sample period. (If the Turbonomic database has fewer days' data, then it uses all of its stored historical data.)

A shorter period means there are fewer data points to account for when Turbonomic calculates utilization. This results in more dynamic, elastic resizing, while a longer period results in more stable or less elastic resizing. You can make the following settings:

- Less Elastic – Last 30 Days
- Recommended – Last 7 Days
- More Elastic – Last 3 Days

## Pool Utilization

These settings affect the actions Turbonomic recommends as it manages business users and active accounts on the desktop pool. Turbonomic recommends actions that avoid using these resources beyond the given settings.

| Attribute                | Default Value |
|--------------------------|---------------|
| Pool CPU Utilization     | 95            |
| Pool Mem Utilization     | 95            |
| Pool Storage Utilization | 95            |

The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity. For example, setting 70 for Desktop Pool Pool CPU Utilization means that Turbonomic considers 70% utilization of that CPU to be 100% of capacity and 35% utilization to be 50% of capacity.

## Placement Policies

Under some circumstances, you can have Business Users who need larger desktop images. This appears as users with high utilization of the image resources. In this case, Turbonomic can recommend moving the Business Users to a different desktop pool that serves up larger images.

To support moving Business Users, you must create a placement policy that merges desktop pools. Be sure to merge only desktop pools that are *similarly configured* – they should run the same operating system and applications, and differ only in allocated memory and/or CPU.

To merge desktop pools:

1. Create a new placement policy.
2. Choose **Merge** as the policy type.
3. For the consumer type to merge, choose **Desktop Pool**.
4. Choose the pools that you want to merge.
5. Save the policy.

For more information, see [Creating Placement Policies](#).

## View Pod

For Virtual Desktop Infrastructure (VDI) environments, a View Pod groups together a given set of Desktop Pools.

## Synopsis



### Synopsis

|                     |                                                                                                  |
|---------------------|--------------------------------------------------------------------------------------------------|
| Budget:             | A View Pod has unlimited budget.                                                                 |
| Provides:           | Active Sessions.                                                                                 |
| Consumes:           | N/A                                                                                              |
| Discovered through: | The VDI management target.<br>For VMware Horizon View, the target is the View Connection Server. |

### Monitored Resources

Turbonomic monitors the following resources for a Desktop Pool:

- Active Sessions

How many active sessions are on the pool as a percentage of the pool's capacity as defined in the Turbonomic policy.

- Total Sessions

How many active and disconnected (non-terminated) sessions are on the pool, as a percentage of the pool's capacity.

### Actions

#### None

Turbonomic does not recommend actions for a view pod. Instead, it recommends actions for the Business Users that are running active sessions.

### Active Session Capacity for View Pods

Each View Pod entity has a set capacity of active sessions. By default, Turbonomic assumes a capacity of 8,000. So that Turbonomic can generate reliable actions for Business User entities, you must set this capacity to match the active session capacity that your Horizon administrator has deployed for the given view pod.



Once you know the correct active session capacity for your view pod, create an automation policy that sets the capacity. For complete information about creating automation policies, see [Creating Automation Policies](#). For information about view pod policies, see [View Pod Policies](#).

1. Create a new automation policy.

Navigate to the Settings Page and choose **Policies**. Then click **NEW AUTOMATION POLICY**, and select View Pod as the policy type. Be sure to name the new policy.

2. Set the policy scope to your view pod.

To define its scope, you assign a group to the policy. You will have to create the group for this view pod:

- Expand the **SCOPE** section and then click **ADD VIEW POD GROUPS**.
- Choose the group that contains only the view pod you want to configure.

If it has already been created, choose the group from the list. If the group does not appear, click **NEW GROUP** to create a static group that includes only the view pod you want to configure. For more information about creating groups, see [Creating Groups](#).

Choose the group you want and click **SELECT**. This returns you to the Configure View Pod Policy fly-out.

3. Set the view pod capacity.

Expand the **UTILIZATION CONSTRAINTS** section and click **ADD UTILIZATION CONSTRAINT**. From the drop-down list, choose Active Sessions Capacity. In the capacity field, enter the capacity that you have calculated for your desktop pools.

4. Save your work

When you're done, be sure to click **SAVE AND APPLY**.

## View Pod Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

#### None

Turbonomic does not recommend actions for a view pod. Instead, it recommends actions for the Business Users that are running active sessions.

### Active Sessions Capacity

This setting controls the number of active sessions a given view pod can support.

| Attribute                | Default Value |
|--------------------------|---------------|
| Active Sessions Capacity | 8000          |

For each view pod, you should set this value to match the active session capacity that has been deployed in your VDI environment for the given view pod. For more information, see [Active Session Capacity for View Pods](#).

## Host

For on-prem environments, a host is a server that runs processes, including hypervisor processes to host virtual workloads. Note that a host is not necessarily a physical piece of hardware. A VM can be set up as

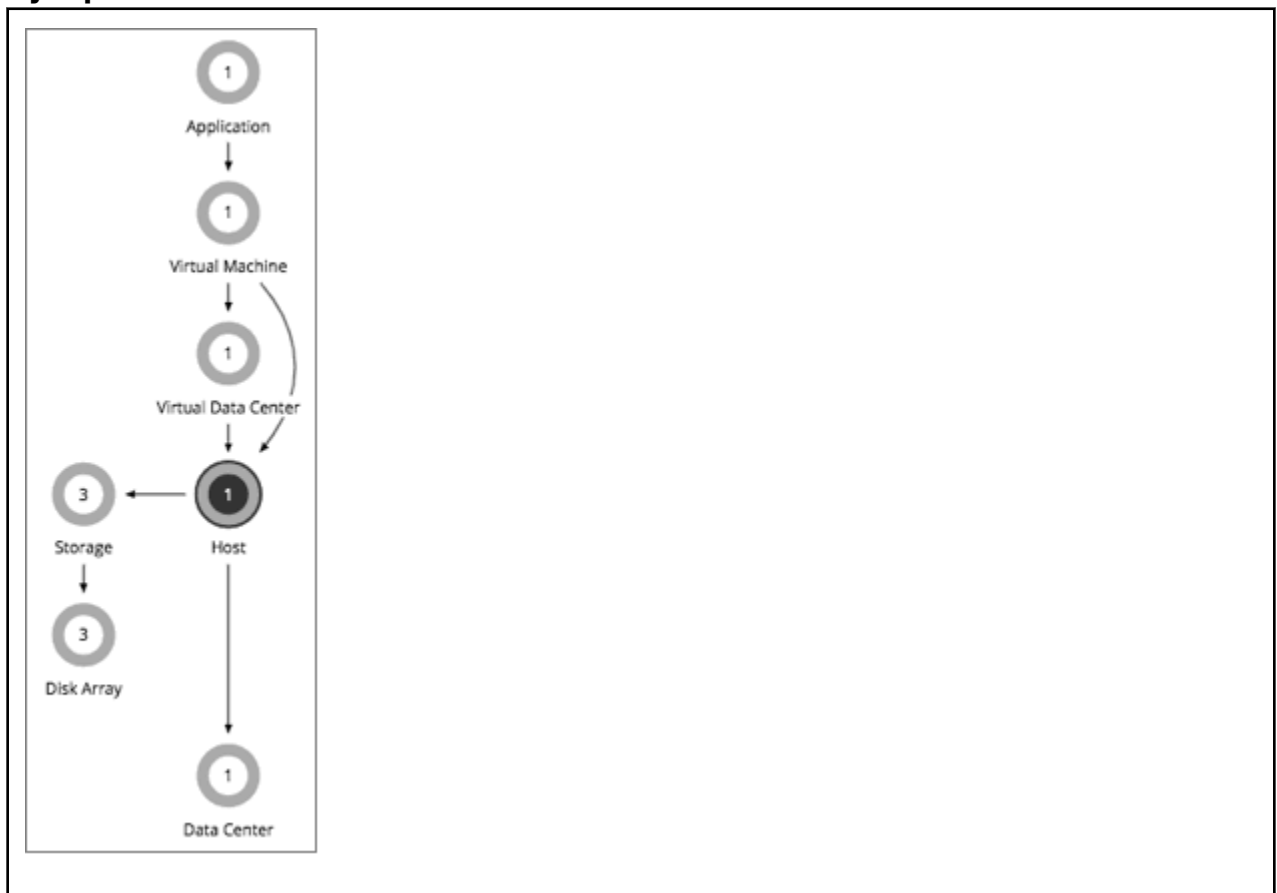
a server that runs a hypervisor, and in turn it can host other VMs within its processing space. However, it's most usual to use physical hardware as your hosts.

**Note:**

To support vSAN storage in your environment, you can deploy HCI Hosts. Turbonomic discovers the vSAN as a storage entity that consumes resources from the underlying hosts. For more information, see [vSAN Storage](#).

On the public cloud a host is an availability zone. This is where your cloud workloads run. For details, see [Zone](#).

**Synopsis**



**Synopsis**

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget: | A host gains its budget by selling resources to the workloads that run on it. The more workloads running on a host, the more budget the host has to purchase storage and datacenter resources. If utilization of a host is high enough, Turbonomic can recommend that you provision a new one. If utilization falls off, the host loses budget. Ultimately, if the budget isn't enough to pay for the services it consumes, Turbonomic will recommend to suspend or power off the host. |
|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Synopsis

|                     |                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Provides:           | Host resources for VMs to use: <ul style="list-style-type: none"><li>• Mem (Kbytes)</li><li>• CPU (MHz)</li><li>• IO (throughput on the I/O bus)</li><li>• Net (network throughput)</li><li>• Swap (swap rate capacity measured in bytes/sec)</li><li>• Ballooning (sharing of memory among hosted VMs)</li><li>• CPU Ready Queue (wait time on the queue in ms)</li></ul> |
| Consumes:           | Datacenter resources (physical space, cooling, etc.) and storage.                                                                                                                                                                                                                                                                                                          |
| Discovered through: | Turbonomic discovers hosts through hypervisor targets. For some hypervisor vendors, the host is the target, and for others the hosts are managed by the specified target.                                                                                                                                                                                                  |

## Monitored Resources

Turbonomic monitors the following resources on a host:

- Memory (Mem)

The utilization of the host's memory reserved or in use

- CPU

The utilization of the host's CPU reserved or in use

- IO

The utilization of the host's IO adapters

- Net

The utilization of data through the host's network adapters

- Swap

The utilization of the host's swap space

- Balloon

The utilization of shared memory among VMs running on the host. ESX-only

- CPU Ready

The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only

For hosts discovered via vCenter, HPE OneView, and UCS Manager targets, the following resources are also monitored:

- Energy

Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

- Carbon Footprint

Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

## Actions

- **Start**

Start a suspended host when there is increased demand for physical resources.

- **Provision**

Provision a new host in the environment when there is increased demand for physical resources. Turbonomic can then move workloads to that host.

- **Suspend**

When physical resources are underutilized on a host, move existing workloads to other hosts and then suspend the host.

- **Reconfigure**

Turbonomic generates this action in response to changing demand for software licenses. For details, see [License Policy](#).

**Note:**

Turbonomic discovers VMware HA configurations in clusters, and considers the reserved resources in its calculations. For tolerated host failures, or a reserved percentage of cluster resources, Turbonomic automatically sets utilization constraints for that cluster. If you configure a failover host, Turbonomic reserves that host for HA and will not move VMs to it.

## DRS Automation Settings

Turbonomic automatically discovers DRS automation settings for vSphere hosts managed through vCenter. When you set the scope to a vSphere host and then view the Entity Information chart, the following information displays:

- Vendor Automation Mode

The chart shows the automation mode discovered from vCenter – Not Automated, Partially Automated, or Fully Automated.

- Vendor Migration Level

Turbonomic assigns a vendor migration level based on the migration level discovered from vCenter. The chart only shows the assigned migration level (i.e., the Turbonomic Vendor Migration Level).













| <b>Turbonomic Vendor Migration Level</b> | <b>vCenter Migration Level</b> |
|------------------------------------------|--------------------------------|
| 1 (Conservative)                         | 5                              |
| 2 (Less Conservative)                    | 4                              |
| 3 (Moderate)                             | 3                              |
| 4 (Less Aggressive)                      | 2                              |
| 5 (Aggressive)                           | 1                              |

## Host Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For details about host actions, see [Host Actions](#).

| Action      | Default Mode | vCenter                                                                           | Hyper-V                                                                           | UCS (blades only)                                                                   | PowerVM |
|-------------|--------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------|
| Start       | Recommend    |  |  |  | N/A     |
| Suspend     | Recommend    |  |  |  | N/A     |
| Provision   | Recommend    |  |  |  | N/A     |
| Reconfigure | Recommend    |  |  |  | N/A     |

You can use Action Scripts for action orchestration.

For ServiceNow:

- Host provision actions will not generate a CR.
- For host suspend actions to succeed, it must be enabled in the given hypervisor, and there must be no VMs currently running on that host.

## Maintenance Automation Avoidance

| Attribute                        | Default Setting |
|----------------------------------|-----------------|
| Maintenance Automation Avoidance | 30 minutes      |

The Maintenance Automation Avoidance setting applies to vCenter environments with DRS clusters. Turbonomic uses this setting when:

- Turbonomic actions to move VMs from one host to another are automated.
- The DRS [automation level](#) is *Fully Automated*, regardless of [migration threshold](#).

### Note:

Turbonomic automatically discovers DRS automation levels and migration thresholds and displays them in the Entity Information chart for hosts.

- Host maintenance is in effect.

This setting prevents action conflicts between Turbonomic and DRS.

When a host enters maintenance mode, DRS starts to move VMs on the host to other hosts to prepare for maintenance. In response, Turbonomic clears all pending actions to and from the host. For example, assume a cluster with Host\_01, Host\_02, and Host\_03. When Host\_01 enters maintenance mode, Turbonomic removes the following pending actions from the system:

- Move a VM on Host\_01 to Host\_02.

This prevents a potential conflict with a DRS action that moves the VM to Host\_03.

- Move a VM on Host\_02 to Host\_01.

Since Host\_02 and Host\_03 are not in maintenance mode, Turbonomic might recommend moving the VM from Host\_02 to Host\_03 as an alternative action.

In addition, Turbonomic treats a host entering or in maintenance mode as uncontrollable and stops generating actions for the host. The host remains uncontrollable after it leaves maintenance mode, but is within the Maintenance Automation Avoidance period that you specified. During rolling host maintenance operations on DRS clusters, where hosts undergo maintenance on a staggered basis, this gives DRS a window (30 minutes by default) to move VMs from hosts entering maintenance to hosts that have recently left maintenance, thereby avoiding any potential conflict.

When the Maintenance Automation Avoidance period is over, Turbonomic treats the host as controllable and resumes action generation. At this stage, it is assumed that all critical DRS activities on the host have been completed, so Turbonomic actions should be safe to execute.

The following table summarizes Turbonomic's response at various stages of maintenance.

| Maintenance Status                                                                         | DRS Activities                                                                           | Host Status in Turbonomic         | Turbonomic Pending Actions   | Turbonomic New Actions |
|--------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|-----------------------------------|------------------------------|------------------------|
| Host is entering maintenance mode.                                                         | Increased number of DRS activities moving VMs away from the host entering maintenance    | X<br>Uncontrollable (Maintenance) | ✓<br>Removed from the system | X<br>Not generated     |
| Host is in maintenance mode.                                                               | Maintenance tasks on the host                                                            | X<br>Uncontrollable (Maintenance) | N/A                          | X<br>Not generated     |
| Host has left maintenance mode but is within the Maintenance Automation Avoidance window.  | Increased number of DRS activities moving VMs away from other hosts entering maintenance | X<br>Uncontrollable (Maintenance) | N/A                          | X<br>Not generated     |
| Host has left maintenance mode and is outside the Maintenance Automation Avoidance window. | Minimal number of DRS activities on the host                                             | ✓<br>Controllable                 | N/A                          | ✓<br>Generated         |

**Points to consider:**

- You can set a different Maintenance Automation Avoidance value that aligns with your host maintenance practices. For example, if moving VMs back to a host typically takes an hour, specify a value of 60.
- You can set a global value in the default policy for hosts, or specific values in automation policies that you create for your clusters.
- For rolling maintenance of hosts in a cluster, where hosts undergo maintenance on a staggered basis, there could be a point in the process where some or all hosts are uncontrollable. This means that Turbonomic cannot recommend actions to alleviate pressure on overburdened hosts. As such, these hosts could lose performance while they are uncontrollable.
- This setting has no effect on clusters where the DRS automation level is *Manual* or *Partially Automated*. As soon as a host enters maintenance mode, Turbonomic automates the first action to move a VM to another host, and then stops recommending actions. After the host leaves maintenance mode, Turbonomic automates actions to manage the performance of the cluster as normal.

## Utilization Constraints

Utilization constraints affect the actions Turbonomic recommends as it manages your environment. Turbonomic recommends actions that avoid using these resources beyond the given settings. The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute                         | Default Value |
|-----------------------------------|---------------|
| Memory Overprovisioned Percentage | 1000          |
| Net Throughput                    | 50            |
| Ready Queue Utilization           | 50            |
| Memory Utilization                | 100           |
| IO Throughput                     | 50            |
| CPU Overprovisioned Percentage    | 1000          |
| CPU Utilization                   | 100           |
| Swapping Utilization              | 20            |

For example:

- Setting 50 for Net Throughput means that Turbonomic considers 50% utilization of that throughput to be 100% of capacity and 25% utilization to be 50% of capacity
- Setting 1000 for Memory Overprovisioned Percentage means that overprovisioning memory by 5 times the physical capacity shows up as 50% utilization of the Mem Overprovisioned capacity in Turbonomic
- Setting 100 for Memory Utilization means that Turbonomic capacity reflects the physical capacity for this resource

## Desired State

The desired state for your environment is an n-dimensional sphere that encompasses the fittest conditions your environment can achieve.

| Attribute | Default Value |
|-----------|---------------|
| Diameter  | 10            |
| Center    | 70            |

The multiple dimensions of this sphere are defined by the resource metrics in your environment. Metric dimensions include VMem, storage, CPU, etc. While the metrics on the devices in your environment can be any value, the desired state, this n-dimensional sphere, is the subset of metric values that assures the best performance while achieving the most efficient utilization of resources that is possible.

The Desired State settings define the center of the sphere as well as its diameter. This is a way for you to customize what Turbonomic considers to be the desired state.

Setting the center of the sphere chooses the priority for Turbonomic analysis. If you set the balance in favor of efficiency, Turbonomic tends to place more VMs on fewer physical hosts, and to give them storage capacity from fewer data stores. As a result, high utilization can have more impact on QoS. With a balance

in favor of performance, Turbonomic tends to spread virtual loads across more physical devices. This can result in the provisioning of excess resources.

The diameter setting determines the range of deviation from the center that can encompass the desired state. If you specify a large diameter, Turbonomic will have more variation in the way it distributes workload across hosting devices.

As you move each slider, a tooltip displays the numerical value of the setting. **Center** indicates the percentage of resource utilization you want, within the range you specify as **Diameter**. For example, if you want utilization of 75%, plus or minus 10%, then you would set **Center** = 75 and **Diameter** = 20. Turbonomic recommends actions that tend toward this desired state much as possible, given the dependencies within the current environment.

**Note:**

The setting for Target Utilization can have an effect on plans that you run. If you disable provisioning and suspension for hosts and datastores, then you should always set Center and Diameter to their default values.

### Placement Policies

You can create placement policies that merge multiple clusters into a single logical group for the purpose of workload placement.

For example, you can merge three host clusters in a single provider group. This enables Turbonomic to move workload from a host in one of the clusters to a host in any of the merged clusters to increase efficiency in your environment.

For more information, see [Creating Placement Policies](#).

**Note:**

For vCenter, IBM automatically imports any vSphere Host DRS rules when DRS is enabled, and displays them on the **Settings > Policies** page under **Imported Placement Policies**.

For more information, see [Importing Workload Placement Policies](#).

## Chassis

A chassis houses the servers that are part of a computing fabric. It provides compute, memory, storage, and bandwidth resources.

### Synopsis



| Synopsis  |                                                    |
|-----------|----------------------------------------------------|
| Budget:   | A Chassis has unlimited budget.                    |
| Provides: | Chassis resources (physical space, cooling, etc.). |



## Synopsis

Consumes:

N/A

Discovered through:

Turbonomic discovers Chassis through fabric manager targets.

### Note:

When Turbonomic discovers that blade servers housed in a particular chassis have been designated as vCenter hosts, the supply chain stitches the blade servers and chassis to the corresponding vCenter data center to establish their relationship. When you set the scope to that data center and view the Health chart, you will see the blade servers in the list of hosts. In addition, when the data center is included in a merge policy (a policy that merges data centers for the purpose of VM placement), the VMs in the blade servers apply the policy, allowing them to move between data centers as necessary.

### Monitored Resources

Turbonomic monitors the following resources for the servers in a chassis:

- Power

Electricity being consumed by the Chassis.

- Cooling

The percentage of the acceptable temperature range that is utilized by this chassis. As the chassis temperature nears the high or low running temperature limits, this percentage increases.

### Actions

#### None

Turbonomic does not recommend actions for a chassis.

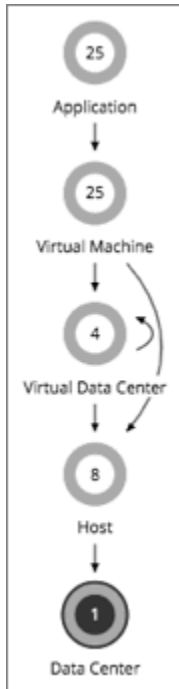
## Data Center

A data center is the sum of VMs, hosts, datastores, and network devices that are managed by a given hypervisor target. A data center provides compute, memory, storage, and bandwidth resources.

### Note:

For public cloud environments, a data center is the cloud region. The hosts that get resources from the data center are availability zones within that region. For details, see [Region](#) and [Zone](#).

## Synopsis



### Synopsis

|                     |                                                               |
|---------------------|---------------------------------------------------------------|
| Provides:           | Compute, memory, storage, and bandwidth resources             |
| Consumes:           | N/A                                                           |
| Discovered through: | Turbonomic discovers data centers through hypervisor targets. |

### Note:

When Turbonomic discovers that blade servers housed in a particular chassis have been designated as vCenter hosts, the supply chain stitches the blade servers and chassis to the corresponding vCenter data center to establish their relationship. When you set the scope to that data center and view the Health chart, you will see the blade servers in the list of hosts. In addition, when the data center is included in a merge policy (a policy that merges data centers for the purpose of VM placement), the VMs in the blade servers apply the policy, allowing them to move between data centers as necessary.

### Monitored Resources

Turbonomic does not monitor resources directly from the data center, but it does monitor the following resources, aggregated for the hosts in a data center:

- Memory (Mem)  
The utilization of the host's memory reserved or in use.
- CPU  
The utilization of the host's CPU reserved or in use.
- IO  
The utilization of the host's IO adapters.
- Net

The utilization of data through the host's network adapters.

- Swap

The utilization of the host's swap space.

- Balloon

The utilization of shared of memory among VMs running on the host. ESX-only.

- CPU Ready

The utilization of the host's allocated ready queue capacity that is in use, for 1, 2, and 4 CPU ready queues. ESX-only .

For data centers with hosts discovered via vCenter, HPE OneView, and UCS Manager targets, the following resources are also monitored:

- Energy

Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

- Carbon Footprint

Carbon footprint is the measurement of carbon dioxide equivalent (CO<sub>2</sub>e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

## Actions

### None

Turbonomic does not recommend actions for a data center. Instead, it recommends actions for the entities running in the data center.

## Data Center Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Operational Constraints

Turbonomic calculates carbon footprint using industry standards that take into account energy consumption, datacenter efficiency, and carbon intensity data. You can create Data Center policies to adjust the calculations according to the requirements of your data centers. For example, a data center in a particular location might have different requirements than data centers in other locations. After you adjust the calculations via policies, Turbonomic can accurately report your organization's carbon footprint.

| Attribute                       | Default Value |
|---------------------------------|---------------|
| Carbon Intensity (CI)           | .25 (g/Wh)    |
| Power Usage Effectiveness (PUE) | 1.5           |

- Carbon Intensity (CI) is a measurement of how 'clean' electricity is. It refers to how many grams of carbon dioxide (CO<sub>2</sub>) are released to produce 1 watt-hour (Wh) of electricity. Electricity that is generated using fossil fuels is more carbon intensive, as the process by which it is generated creates CO<sub>2</sub> emissions. Renewable energy sources, such as wind, hydro, or solar power produce next to no CO<sub>2</sub> emissions, so their carbon intensity value is much lower and often zero.
- Power Usage Effectiveness (PUE) is a ratio that describes how efficiently a computer data center uses energy; specifically, how much energy is used by the computing equipment. PUE is the ratio of the total amount of energy used by a computer data center facility to the energy delivered to computing equipment. The closer PUE is to 1, the more efficient the computer data center.

## Placement Policies

For vCenter environments, you can create placement policies that merge data centers to support cross-vCenter moves. In this case, where a data center corresponds to a given vCenter target, the merged clusters can be in different data centers. In this case you must create two merge policies; one to merge the affected data centers, and another to merge the specific clusters.

For more information, see [Creating Placement Policies](#).

## Storage

Turbonomic represents storage as Datastores. A Datastore is a logical grouping of one or more physical storage devices that serve workload storage requirements.

### Synopsis



| Synopsis            |                                                                                                                                                                                                               |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | A Datastore gains its budget by selling resources to the VMs it serves. If utilization of a Datastore is high enough, Turbonomic can recommend that you provision a new one.                                  |
| Provides:           | Host resources for VMs to use: <ul style="list-style-type: none"> <li>• Storage amount</li> <li>• IOPS (storage access operations per second)</li> <li>• Latency (capacity for disk latency in ms)</li> </ul> |
| Consumes:           | Disk arrays (or aggregates)                                                                                                                                                                                   |
| Discovered through: | Turbonomic discovers on-prem Datastores through hypervisor targets and storage controllers.                                                                                                                   |

### Monitored Resources

Turbonomic monitors the following resources for a datastore:

- Storage Amount
  - The utilization of the datastore's capacity.
- Storage Provisioned
  - The utilization of the datastore's capacity, including overprovisioning.

- Storage Access Operations Per Second (IOPS)

The summation of the read and write access operations per second on the datastore.

**Note:**

When it generates actions, Turbonomic does not consider IOPS throttling that it discovers on storage entities. Analysis uses the IOPS it discovers on Logical Pool or Disk Array entities.

- Latency

The utilization of latency on the datastore.

## Storage Actions

- **Move**

For high utilization of physical storage, move datastore to a different disk array (aggregate).

- **Provision**

For high utilization of storage resources, provision a new datastore.

- **Resize**

Increase or decrease the datastore capacity.

- **Start**

For high utilization of storage resources, start a suspended datastore.

- **Suspend**

For low utilization of storage resources, move served VMs to other datastores and suspend this one.

- **Delete**

Delete a datastore or volume that has been suspended for a period of time.

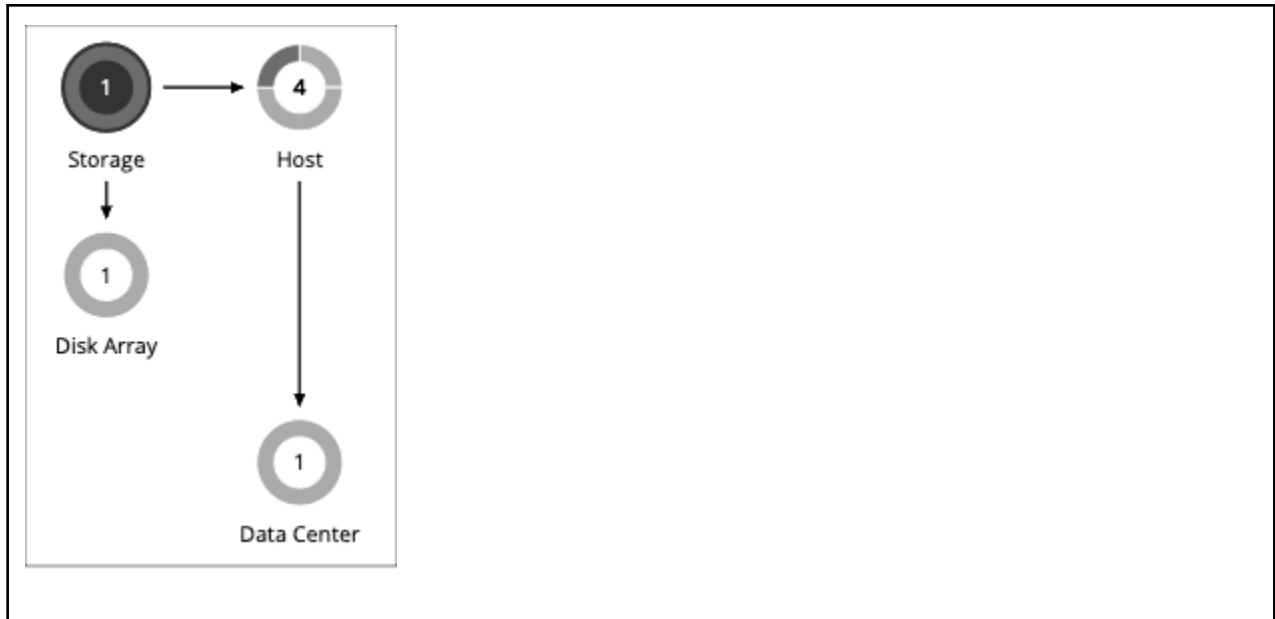
Storage resize actions use Turbonomic tuned scaling settings. This gives you increased control over the action acceptance mode Turbonomic will use for the affected actions. For an overview of tuned scaling, see [Tuned Scaling for On-prem VMs](#).

You can create placement policies to enforce constraints for storage move actions. For example, you can have a policy that allows storage to only move to certain disk arrays, or a policy that prevents storage from moving to certain disk arrays.

For more information, see [Creating Placement Policies](#).

## vSAN Storage

### Overview



For environments that use hyperconverged infrastructure to provide storage on a vSAN, Turbonomic can discover the storage provided by a host cluster as a single Storage entity. This Storage entity represents the full storage capacity that is provided by that host cluster.

Turbonomic supports VMware vSAN, but does not support stretched VSAN clusters. Adding stretched clusters can cause the generation of incorrect storage recommendations and actions.

Turbonomic supports VMware vSAN.

### vSAN Storage Capacity

When you consider vSAN capacity, you need to compare *Raw Capacity* with *Usable Capacity*.

- Raw Capacity
- Turbonomic discovers Raw Capacity configured in vCenter and uses it to calculate Usable Capacity. Raw Capacity displays in the Entity Information chart.
- Usable Capacity

Turbonomic calculates Usable Capacity and then uses the calculated value to drive scaling actions. Turbonomic can recommend scaling the Storage Amount, Storage Provisioned, or Storage Access capacity. Usable Capacity displays in the Capacity and Usage chart.

### Usable Capacity Calculation

To calculate Usable Capacity, Turbonomic considers a variety of attributes, including:

- Raw Capacity and Largest Host Capacity

Turbonomic compares the Raw Capacity for all the hosts in the cluster and then uses the largest value as Largest Host Capacity.

- RAID Factor

Turbonomic calculates RAID Factor based on the *Failures to Tolerate* (FTT) value and *Redundancy Method* that it discovers. FTT specifies how many failures a given cluster can tolerate, while Redundancy Method specifies the RAID level for the cluster.

| FTT | Redundancy Method | RAID Factor |
|-----|-------------------|-------------|
| 0   | RAID1             | 1           |
| 1   | RAID1             | 1/2         |
| 2   | RAID1             | 1/3         |
| 1   | RAID5/6           | 3/4         |
| 2   | RAID5/6           | 2/3         |

**Note:**

If discovery fails for some reason, Turbonomic uses a RAID Factor of 1.

- Host Capacity Reservation, Slack Space Percentage, and Compression Ratio

You can control the values for these attributes in storage policies. For details about these attributes and their effect on usable capacity calculations, see [Hyper-converged Infrastructure Settings](#).

The calculation for Usable Capacity can be expressed as:

$$\text{Usable Capacity} = (\text{Raw Capacity} - \text{Largest Host Capacity} * \text{Host Capacity Reservation}) * \text{Slack Space Percentage} * \text{RAID Factor} * \text{Compression Ratio}$$

If the result of the calculation is zero or a negative value, Turbonomic sets the Usable Capacity to 1 MB.

### Capacity and Usage Chart for vSAN Storage

The **Capacity and Usage** chart for vSAN storage shows two Storage Amounts - *Consumed* (bought) and *Provided* (sold). This is because vSAN storage can buy and sell commodities to hosts.

For the *Provided* Storage Amount, the *Capacity* value corresponds to *Usable Capacity*, while the *Used* value indicates utilization.

### Entity Information Chart for vSAN Storage

The **Entity Information** chart includes the following information:

- HCI Technology Type

The technology that supports this storage cluster. For this release, Turbonomic supports VMware vSAN technology.

- Capacity

Turbonomic displays rounded values for the following, which might be slightly different from the values it discovers from vCenter:

- Raw Capacity

The sum of the Raw Capacity that each storage capacity device provides.

- Raw Free Space

How much of the Raw Capacity is not currently in use.

- Raw Uncommitted Space

In terms of Raw Capacity, how much space is available according to your thin/thick provisioning.

- Redundancy Method and Failures to Tolerate

Redundancy Method specifies the RAID level employed for the cluster. RAID level impacts how much Usable Capacity you can see for a given Raw Capacity. You can use a RAID calculator to determine how the RAID level impacts your Usable Capacity.

Failures to Tolerate specifies how many capacity device failures a given cluster can tolerate. In practical terms, this means how many hosts can come down at the same time, without affecting storage. This value should match the RAID level.

## Actions to Add vSAN Capacity

To scale up storage amount, you add additional hosts that are configured to include their storage in the vSAN array.

When you scope the session to the vSAN storage, you can see actions to scale:

- Storage Amount
- Storage Provisioned
- Storage Access

The action to scale up the storage indicates the amount of storage you need to add. It appears as a recommended action. In fact, to add storage you must add a new host.

When you scope the session to hosts that provide the capacity devices to the storage, you can see the following actions that are related to scaling up the storage capacity:

- Scale up StorageAmount for Storage [MyVsanStorageCluster]
- Provision Host [VSAN\_HostName]

The action to provision a host includes details about the storage cluster. Because you need to manually add hosts to your on-prem environment, this appears as a recommended action.

## Planning With vSAN Storage

For *Hardware Replace* and *Custom* plans, you can use HCI Host templates to add vSAN capacity. These represent the hosts that add storage capacity to a vSAN cluster. For more information, see [HCI Host Template Settings](#).

Under certain circumstances, *Add Virtual Machines* plans can fail to place workloads, or it can fail to generate actions to increase storage capacity by provisioning new hosts.

- If you scope the plan to a user-created group that only provides vSAN storage, or to a discovered storage cluster group, then the plan can fail to place VMs with multiple volumes. This can occur for VMs that use conventional storage (not vSAN) along with vSAN storage.
- If you scope the plan to a vSAN host group and add VMs, the plan can fail to increase storage capacity by provisioning new hosts. For example, assume you scope the plan to a vSAN host group and add 20 VMs to the environment. In that case, you need hosts to provide compute capacity for the VMs, and you also need hosts to provide storage capacity. The plan can represent the compute provisioning correctly, but it can incorrectly fail to add more storage capacity to the vSAN.
- If the vSAN RAID type is Raid6/FTT=2, if you scope the plan to any vSAN groups then the plan will fail to place any of the VMs.

## Storage Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

The following are the storage actions and automation support for environments that do not include Disk Array Storage Controllers as targets. For details about these actions, see [Storage Actions](#).



| Action                                                           | Default Mode | vCenter | Hyper-V |
|------------------------------------------------------------------|--------------|---------|---------|
| Delete (Volume)                                                  | Recommend    |         |         |
| Suspend                                                          | Manual       |         |         |
| Delete (Datastore)                                               | Disabled     |         |         |
| Move                                                             | Recommend    |         |         |
| Provision                                                        | Recommend    |         |         |
| Start                                                            | Recommend    |         |         |
| Resize (Up, Down, Above Max, or Below Min - using tuned scaling) | Recommend    |         |         |

For datastores on disk arrays:

| Action             | Default Mode | Dell Compellent | HP 3Par       | NetApp ONTAP | VNX | VMAX | Nutanix | Pure Storage |
|--------------------|--------------|-----------------|---------------|--------------|-----|------|---------|--------------|
| Delete (Volume)    | Recommend    |                 | Not supported |              |     |      |         |              |
| Suspend            | Manual       |                 |               |              |     |      |         |              |
| Delete (Datastore) | Disabled     |                 | Not supported |              |     |      |         |              |
| Move               | Recommend    |                 | Not supported |              |     |      |         |              |
| Provision          | Recommend    |                 |               |              |     |      |         |              |
| Start              | Recommend    |                 |               |              |     |      |         |              |

| Action                                                           | Default Mode | Dell Compellent | HP 3Par | NetApp ONTAP | VNX  | VMAX | Nutanix | Pure Storage |
|------------------------------------------------------------------|--------------|-----------------|---------|--------------|------|------|---------|--------------|
| Resize (Up, Down, Above Max, or Below Min - using tuned scaling) | Recommend    | Rcmd            | Rcmd    | Rcmd         | Rcmd | Rcmd | Rcmd    | Rcmd         |

You can use Action Scripts for action orchestration.

For ServiceNow:

- Storage suspend and vSAN storage resize actions will not generate a CR.
- Currently Turbonomic can only execute a CR for storage provision actions on Pure and Dell Compellent storage.

## Utilization Constraints

Utilization constraints affect the actions Turbonomic recommends as it manages your environment. Turbonomic recommends actions that avoid using these resources beyond the given settings. The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute                       | Default Value |
|---------------------------------|---------------|
| Storage Amount Utilization      | 90            |
| Storage Provisioned Utilization | 100           |
| IOPS Utilization                | 100           |
| Latency Utilization             | 100           |

For example, setting 90 for Storage Amount Utilization means that Turbonomic considers 90% utilization of the physical storage to be 100% of capacity.

## Storage Settings

| Attribute                          | Default Setting/Value |
|------------------------------------|-----------------------|
| Storage Overprovisioned Percentage | 200                   |
| IOPS Capacity                      | 50000                 |
| Storage Latency Capacity [ms]      | 100                   |
| Minimum Wasted Files [KB]          | 1000                  |

| Attribute             | Default Setting/Value                                                      |
|-----------------------|----------------------------------------------------------------------------|
| Directories to Ignore | \\.dvsData.* \\.snapshot.* \\.vSphere-HA.* \\.naa.* \\.etc.* lost +found.* |
| Files to Ignore       | Empty String                                                               |

- **Storage Overprovisioned Percentage**

Storage Overprovisioned Percentage sets how much overprovisioning Turbonomic assumes when recommending actions for VM datastores. For example, if a datastore has a 30 GB capacity, and Storage Overprovisioned Percentage is set to 200, Turbonomic will treat the datastore as though it has a capacity of 60 GB, or 200% of the actual datastore capacity.

- **IOPS Capacity**

IOPS Capacity is the IOPS setting for individual datastores. To set a specific capacity for one group of datastores, select that group as the property scope and override the global setting for that scope.

Note that IOPS capacity for a disk array takes precedence — Datastores that are members of a disk array always have the IOPS capacity that is set to the disk array.

Turbonomic considers these settings when calculating utilization percentage. For example, assume IOPS Capacity of 500 for datastores. If utilization on a datastore is 250 IOPS, then the datastore is at 50% of capacity for that metric.

- **Storage Latency Capacity**

This sets the maximum storage latency to tolerate on a datastore, in ms. The default setting is 100 ms.

Turbonomic measures the latency experienced by all VMs and hosts that access the datastore. Assume a default setting of 100 ms. If a datastore exhibits latency of 50 ms, then the Turbonomic will show latency utilization of 50%.

For VMAX environments, Turbonomic discovers SLO for storage latency that you set in VMAX and uses it in analysis. However, if you set a higher storage latency value in a Turbonomic policy, analysis will use that value instead.

- **Minimum Wasted Files**

You can make settings to control how Turbonomic tracks and reports on wasted storage in your environment. Wasted storage is any disk space devoted to files that are not required for operations of the devices or applications in your environment. Wasted storage may indicate opportunities for you to free up disk space, and provide more storage capacity to running VMs and applications.

If there are groups of datastores you don't want to track for wasted storage, set the given scope and disable datastore browsing there. If you prefer not to use Turbonomic resources to track wasted storage, leave the global setting checked.

The settings for **Directories to Ignore** and **Files to Ignore** specify directories and files that Turbonomic will not consider when looking for wasted data storage space. Separate items in these lists with the OR bar (“|”).

## Scaling Constraints

### Rate of Resize

Turbonomic uses the Rate of Resize setting to determine how to make storage resize changes in a single action.

| Attribute      | Default Value |
|----------------|---------------|
| Rate of Resize | High (3)      |

- **Low**

Change the value by one increment only.

- **Medium**

Change the value by an increment that is 1/4 of the difference between the current value and the optimal value.

- **High**

Change the value to be the optimal value.

This default value ensures that resizing to the desired state can be achieved in a single action. This is more efficient than smaller, incremental resizes.

### **Increment Constant for Storage Amount**

This setting controls how many GB to add or subtract when resizing the allocation for a datastore.

| <b>Attribute</b>                           | <b>Default Value</b> |
|--------------------------------------------|----------------------|
| Increment Constant for Storage Amount [GB] | 100 GB               |

### **Hyperconverged Infrastructure Settings**

Turbonomic considers these settings when calculating capacity and utilization for hyperconverged environments.

| <b>Attribute</b>                  | <b>Default Setting/Value</b> |
|-----------------------------------|------------------------------|
| Host Capacity Reservation         | 1                            |
| Host IOPS Capacity                | 50000                        |
| Slack Space Percentage            | 25                           |
| Compression Ratio                 | 1                            |
| Usable Space Includes Compression | Off                          |

**Note:**

Turbonomic uses Host Capacity Reservation, Slack Space Percentage, and Compression Ratio to calculate vSAN usable capacity and drive scaling actions. For more information about usable capacity and how it is calculated, see [vSAN Storage](#).

- **Host Capacity Reservation**

When a host must be taken out of service for maintenance, vSphere will evacuate the data from that host and move it to other hosts in the cluster to maintain the integrity of the replication demanded by the storage policy. For this to happen, there must be enough free raw capacity available to accept the data being evacuated.

Turbonomic uses this setting to determine how many hosts worth of capacity it should subtract from the raw capacity amount before calculating usable capacity. This is not the same as redundancy. It does not specify how the array distributes data to maintain integrity.

- **Host IOPS Capacity**

In addition to calculating usable capacity, Turbonomic needs an estimate of datastore IOPS capacity (storage access). Turbonomic uses the value that you set to provide an estimate of effective IOPS

capacity for each host in the cluster. Total IOPS capacity is the number of hosts in the cluster multiplied by Host IOPS Capacity.

- **Slack Space Percentage**

It is recommended that a vSAN datastore never be filled to prevent vSphere from moving objects/files around the cluster to balance the datastore across all the hosts.

Turbonomic reduces usable capacity by the percentage that you set.

- **Compression Ratio**

vSAN supports both deduplication and compression, which may increase the amount of usable capacity on the datastore. Turbonomic does not try to predict the deduplication or compression ratio, but you can choose to include a compression ratio into the usable capacity calculation. This captures the ratio achieved both by compression and deduplication.

The compression ratio that you set acts as a multiplier on the raw capacity to calculate usable capacity. For example, a compression ratio of 2 would double the amount of usable capacity. The default value of 1 means no compression.

- **Usable Space Includes Compression**

Turn this on if you want Turbonomic to consider the compression ratio when calculating storage utilization and capacity. Whether this is on or off, Turbonomic always considers compression when calculating utilization of StorageProvisioned.

## Placement Policies

Turbonomic supports placement policies for storage and storage clusters.

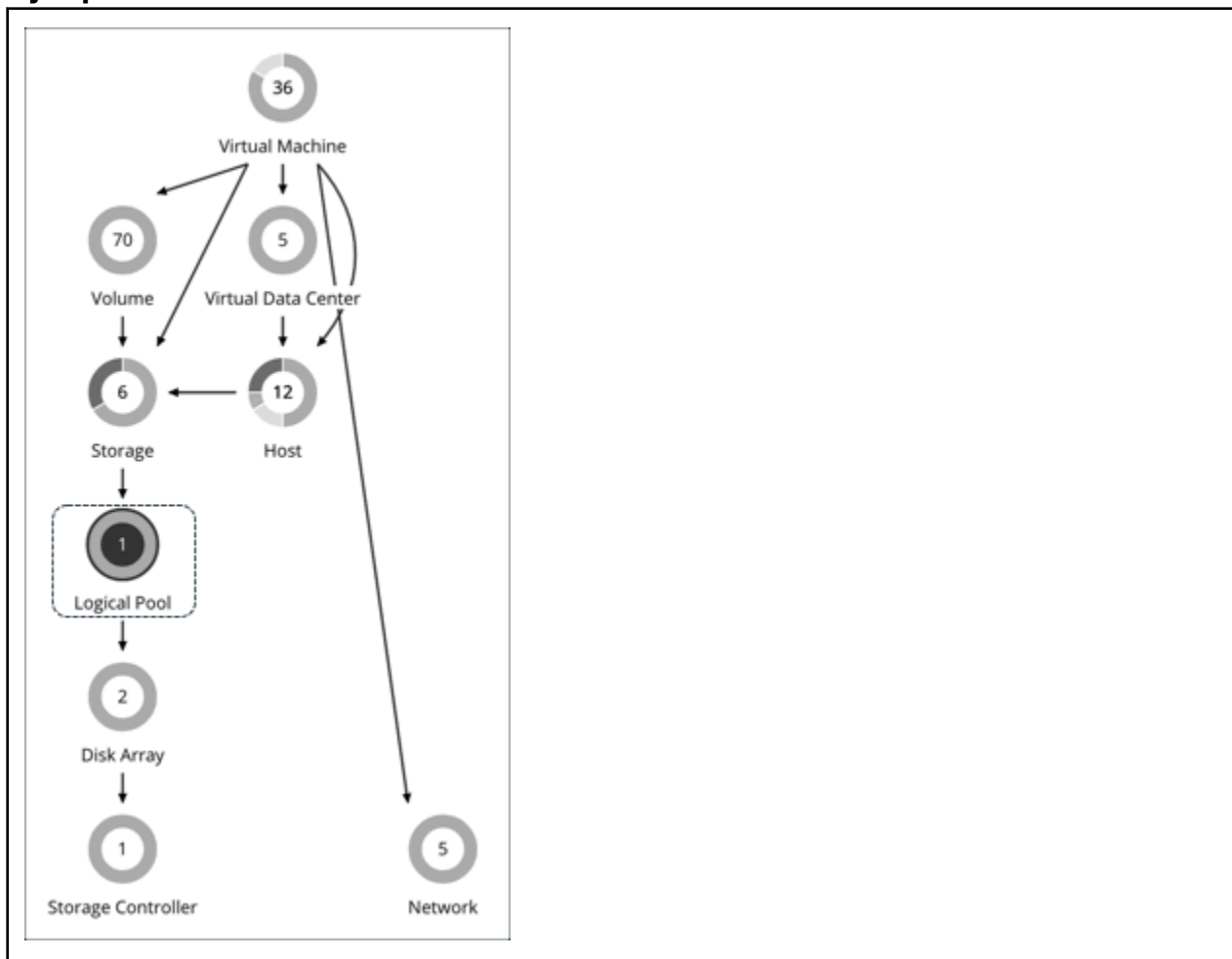
- You can create placement policies to enforce constraints for storage move actions. For example, you can have a policy that allows storage to only move to certain disk arrays, or a policy that prevents storage from moving to certain disk arrays.
- You can create placement policies that merge multiple clusters into a single logical group for the purpose of workload placement.

For more information, see [Creating Placement Policies](#).

## Logical Pool

A logical pool represents storage resources that are managed together and presented as a single storage system. Turbonomic analysis identifies performance and efficiency opportunities for a logical pool. For example, it can recommend moving resources into or out of a logical pool, or aggregating resource capacity within the pool.

## Synopsis



## Synopsis

|                     |                      |
|---------------------|----------------------|
| Budget:             | N/A                  |
| Provides:           | Storage resources    |
| Consumes:           | Disk array resources |
| Discovered through: | Storage targets      |

## Monitored Resources

Turbonomic monitors the following resources for a logical pool:

- Storage Amount  
The utilization of the logical pool's capacity.
- Storage Provisioned  
The utilization of the logical pool's capacity, including overprovisioning.
- Storage Access Operations Per Second (IOPS)  
The summation of the read and write access operations per second on the logical pool.
- Latency

The utilization of latency on the logical pool.

## Logical Pool Actions

- Resize
- Provision
- Move
- Start
- Suspend

## Logical Pool Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

| Action    | Default Mode |
|-----------|--------------|
| Suspend   | Disabled     |
| Start     | Disabled     |
| Resize    | Recommend    |
| Move      | Disabled     |
| Provision | Disabled     |

## Storage Settings

| Attribute                          | Default Value |
|------------------------------------|---------------|
| Storage Latency Capacity [ms]      | 100           |
| Storage Overprovisioned Percentage | 200           |
| IOPS Capacity                      | 50000         |

- **Storage Latency Capacity**

This sets the maximum storage latency to tolerate on a logical pool, in ms. The default setting is 100 ms.

- **Storage Overprovisioned Percentage**

Storage Overprovisioned Percentage sets how much overprovisioning Turbonomic assumes when recommending actions for logical pools.

- **IOPS Capacity**

IOPS Capacity is the IOPS setting for individual logical pools.

**Note:**

Turbonomic discovers storage latency and IOPS capacities that you set in your environment (for example VMAX) and uses them in its analysis. These capacities will be overridden by values that you set in Turbonomic policies.

## Disk Array

A Disk Array (an aggregate) is a data storage system made up of multiple disk drives. For example, a RAID is an aggregate that implements redundancy and other data management features. A disk array provides storage volumes to serve the storage requirements of physical machines. It uses the resources of one storage controller, which manages the disk array operation.

### Synopsis



### Synopsis

|                     |                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | A disk array gains its budget by selling resources to the datastores it serves. If utilization of a disk array is high enough, Turbonomic can recommend that you provision a new one.                                                                  |
| Provides:           | Storage resources for datastores to use: <ul style="list-style-type: none"> <li>• Storage amount</li> <li>• Storage Provisioned</li> <li>• IOPS (storage access operations per second)</li> <li>• Latency (capacity for disk latency in ms)</li> </ul> |
| Consumes:           | Storage controllers                                                                                                                                                                                                                                    |
| Discovered through: | Turbonomic discovers disk arrays through storage controller targets.                                                                                                                                                                                   |



## Monitored Resources

Turbonomic monitors the following resources for a disk array:

### Note:

Not all targets of the same type provide all possible commodities. For example, some storage controllers do not expose CPU activity. When a metric is not collected, its widget in the UI will display no data.

- **Storage Amount**

The utilization of the Disk Array's capacity.

- **Storage Provisioned**

The utilization of the Disk Array's capacity, including overprovisioning.

- **Storage Access Operations Per Second (IOPS)**

The summation of the read and write access operations per second on the disk array.

- **Latency**

The utilization of latency, computed from the latency of each device in the disk array.

## Disk Array Actions

- **Provision**

For high utilization of the disk array's storage, provision a new disk array (recommendation, only).

- **Start**

For high utilization of disk array, start a suspended disk array (recommendation, only).

- **Suspend**

For low utilization of the disk array's storage, move VMs to other datastores and suspend volumes on the disk array (recommendation, only).

- **Move**

(Only for NetApp Cluster-Mode) For high utilization of Storage Controller resources, Turbonomic can move an aggregate to another storage controller. The storage controllers must be running.

For high IOPS or latency, a move is always off of the current disk array. All the volumes on a given disk array show the same IOPS and Latency, so moving to a volume on the same array would not fix these issues.

- **Move VM**

For high utilization of Storage on a volume, Turbonomic can move a VM to another volume. The new volume can be on the current disk array, on some other disk array, or on any other datastore.

For high IOPS or latency, a move is always off of the current disk array. All the volumes on a given disk array show the same IOPS and Latency, so moving to a volume on the same array would not fix these issues.

- **Move Datastore**

To balance utilization of disk array resources, Turbonomic can move a datastore to another array.

## Disk Array Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

The following table describes the default action acceptance modes for disk array actions and automation support for environments that have Disk Array Storage Controllers as targets.

| Action      | Default Mode | Dell Compellent | HP 3Par | NetApp ONTAP | VMAX | VNX  | Nutanix | Pure Storage | XTremIO |
|-------------|--------------|-----------------|---------|--------------|------|------|---------|--------------|---------|
| Move        | Disabled     |                 |         | Rcmd         |      |      |         |              |         |
| Provision   | Recommend    | Rcmd            | Rcmd    | Rcmd         |      |      | Rcmd    |              |         |
| Resize (up) | Recommend    | Rcmd            | Rcmd    | Rcmd         | Rcmd | Rcmd | Rcmd    |              |         |
| Start       | Recommend    |                 |         |              |      |      |         |              |         |
| Suspend     | Disabled     |                 |         |              |      |      |         |              |         |

## Action Automation for NetApp Storage Systems

For NetApp storage systems, the actions Turbonomic can automatically perform depend on the NetApp version you are running, and whether the system is running in cluster mode:

| Automated Action                                                    | Cluster-Mode         |
|---------------------------------------------------------------------|----------------------|
| Move VM between datastores, on the same disk array                  | Yes                  |
| Move VM between datastores on different disk arrays                 | Yes                  |
| Move Datastore between disk arrays on the same storage controller   | Yes                  |
| Move Datastore between disk arrays on different storage controllers | Yes                  |
| Resize Storage                                                      | Yes                  |
| Resize Disk Array                                                   | No – Resize up, only |

In addition, for a system running in Cluster-Mode, Turbonomic can recommend moving an aggregate to another storage controller.

## Utilization Constraints

Utilization constraints affect the actions Turbonomic recommends as it manages your environment. Turbonomic recommends actions that avoid using these resources beyond the given settings. The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute                  | Default Value |
|----------------------------|---------------|
| Storage Amount Utilization | 90            |

## Storage Settings

Set capacity for specific storage resources.

| Attribute                                                                                                    | Default Value |
|--------------------------------------------------------------------------------------------------------------|---------------|
| IOPS Capacity<br>A generic setting for disk array IOPS capacity (see <b>Disk Array IOPS Capacity</b> below). | 5000          |
| VSeries LUN IOPS Capacity                                                                                    | 5000          |
| 7.2k Disk IOPS Capacity                                                                                      | 800           |
| 10k Disk IOPS Capacity                                                                                       | 1200          |
| 15k Disk IOPS Capacity                                                                                       | 1600          |
| SSD Disk IOPS Capacity                                                                                       | 50000         |
| Disk Array IOPS Capacity                                                                                     | 10000         |
| Storage Overprovisioned Percentage                                                                           | 200           |
| Storage Latency Capacity [ms]                                                                                | 100           |

### Note:

Turbonomic discovers storage latency and IOPS capacities that you set in your environment (for example VMAX) and uses them in its analysis. These capacities will be overridden by values that you set in Turbonomic policies.

#### • IOPS Capacity

The capacity of IOPS (IO operations per second) that your storage devices can support. Turbonomic considers these settings when calculating utilization percentage. For example, assume IOPS Capacity of 5000 for a disk array. If utilization on the array is 2500 IOPS, then the disk array is at 50% of capacity for that metric.

Note that the IOPS setting for an array will determine IOPS calculations for all the storage on that array. If you made different IOPS settings for individual datastores hosted by the array, Turbonomic ignores the datastore settings and uses the disk array settings.

#### – Various Disk IOPS Capacity settings (**SSD Disk IOPS**, **7.2k Disk IOPS**, etc)

IOPS capacity settings for the different types of physical drives that are discovered on a disk array. If the storage controller exposes the types of disks in the array, Turbonomic uses multiples of these values to calculate the IOPS capacity of the disk array.

#### – **Disk Array IOPS Capacity**

Some disk arrays do not expose data for their individual disks — This is typical for flash arrays, or arrays that aggregate storage utilization across multiple tiers. Turbonomic uses this setting for the IOPS capacity of such disk arrays. Set it to the global scope to specify IOPS capacity for all disk arrays. To override this setting, set a disk array or group of disk arrays as the property scope, and then set the value you want for **IOPS Capacity**.

### Note:

The user interface shows a disk array entity for any array that is discovered through a valid disk array or storage controller target. It also shows *placeholder* disk arrays for disk arrays that are not discovered through a configured target. For example, you might have disk arrays that Turbonomic does not natively

support. Or you might have storage that is not hosted by any disk array. Such *placeholder* disk array entities appear with the string "DiskArray-" prefixed to their names. The user interface allows you to set IOPS Capacity to these placeholders, but those settings have no effect. To set IOPS Capacity for that storage, you must set it to the individual datastores.

- **Storage Overprovisioned**

This setting indicates how much overprovisioning Turbonomic assumes when recommending actions for disk arrays. For example, if a disk array has a 30 TB capacity, and DiskArray Overprovisioned Percentage is set to 200, Turbonomic will treat the datastore as though it has a capacity of 60 TB, or 200% of the actual disk array capacity.

## Storage Controller

A Storage Controller is a device that manages one or more disk arrays. The storage controller provides CPU cycles to perform storage management tasks for each disk array it manages.

### Synopsis



### Synopsis

|                     |                                                                                                                                                                                                                                                                   |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Budget:             | A storage controller gains its budget by selling resources to the disk arrays it manages. If utilization of the storage controller's CPU resources is high enough, Turbonomic can recommend that you provision a new one and move disk arrays (aggregates) to it. |
| Provides:           | CPU resources to manage disk arrays.                                                                                                                                                                                                                              |
| Consumes:           | NA                                                                                                                                                                                                                                                                |
| Discovered through: | Turbonomic directly accesses storage controller targets.                                                                                                                                                                                                          |

### Monitored Resources

Turbonomic monitors the following resources for a storage controller:

- CPU

The utilization of the Storage Controller's allocated CPU.

- Storage Amount

The utilization of the storage controller's capacity. The storage allocated to a storage controller is the total of all the physical space available to aggregates managed by that storage controller.

**Note:**

In NetApp environments, the storage controller shows 100% utilization when there are no more disks in a SPARE state that the storage controller can utilize in an aggregate. This does not indicate that the storage controller has no capacity.

## Actions

### Provision

For high utilization of the storage controller's CPU, provision a new storage controller, and then move disk arrays to it.

## Storage Controller Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

### Action Automation and Orchestration

Actions for individual Disk Array Storage Controllers:

| Action    | Default Mode | Dell Compellent | HP 3Par | NetApp ONTAP | VNX  | VMAX | Nutanix | Pure Storage | XTremIO |
|-----------|--------------|-----------------|---------|--------------|------|------|---------|--------------|---------|
| Provision | Disabled     | Rcmd            | Rcmd    | Rcmd         | Rcmd | Rcmd | Rcmd    | Rcmd         | Rcmd    |

## Utilization Constraints

Utilization constraints affect the actions Turbonomic recommends as it manages your environment. Turbonomic recommends actions that avoid using these resources beyond the given settings. The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute                  | Default Value                                                                           |
|----------------------------|-----------------------------------------------------------------------------------------|
| Storage Amount Utilization | 90<br>Maximum allowed utilization of storage that is managed by the Storage Controller. |
| CPU Utilization            | 100<br>Maximum allowed utilization of Storage Controller CPU (from 20 to 100).          |

## Storage Settings

Set capacity for specific storage resources.

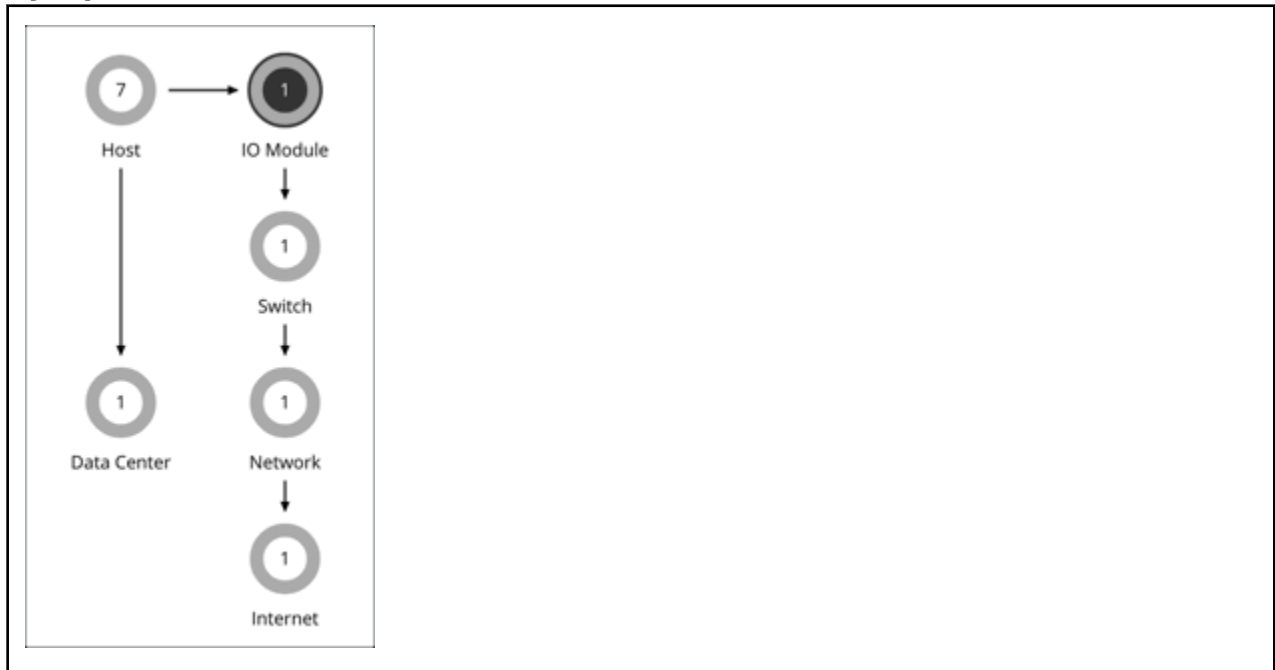
| Attribute                     | Default Value |
|-------------------------------|---------------|
| IOPS Capacity                 | 5000          |
| Storage Latency Capacity [ms] | 100           |

## IO Module

An IO Module connects the compute resources on a chassis to the fabric domain via the Fabric Interconnect. It provides the servers on the chassis with Net resources. Typical installations provide two IO Modules per chassis.

Turbonomic supports IO Modules when you have installed the Fabric Control Module license.

## Synopsis



### Synopsis

|                     |                                                                               |
|---------------------|-------------------------------------------------------------------------------|
| Budget:             | An IO Module gains its budget by selling Net resources to a physical machine. |
| Provides:           | Net resources                                                                 |
| Consumes:           | Chassis and Fabric Interconnect                                               |
| Discovered through: | Turbonomic discovers IO Modules through the fabric managers that use them.    |

### Monitored Resources

Turbonomic monitors the following resources for an IO Module:

- NetThroughput  
Rate of message delivery over a port.

### Actions

#### None

Turbonomic does not recommend actions for an IO Module.

## Switch

A switch connects servers in a computing fabric to the fabric's network and storage resources. It provides network bandwidth to the servers in the platform.

## Synopsis



### Synopsis

|                     |                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------|
| Budget:             | A switch gains its budget by selling Net resources to the IO Modules.                           |
| Provides:           | Net resources                                                                                   |
| Consumes:           | N/A                                                                                             |
| Discovered through: | Turbonomic discovers switches through managers of fabric platforms (such as UCS) that use them. |

### Monitored Resources

Turbonomic monitors the following resources for a switch:

- NetThroughput  
Rate of message delivery over a port.
- PortChannel  
Amalgamation of ports with a shared net throughput and utilization.

### Actions

#### Resize






Resize PortChannel for a switch to increase bandwidth.

### Switch Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

## Action Automation and Orchestration

For environments that have Fabric Managers as targets:

| Action    | Default Mode | Cisco UCS                                                                           |
|-----------|--------------|-------------------------------------------------------------------------------------|
| Resize    | Recommend    |  |
| Start     | Recommend    |  |
| Provision | Recommend    |  |
| Suspend   | Disabled     |  |
| Move      | Disabled     |  |

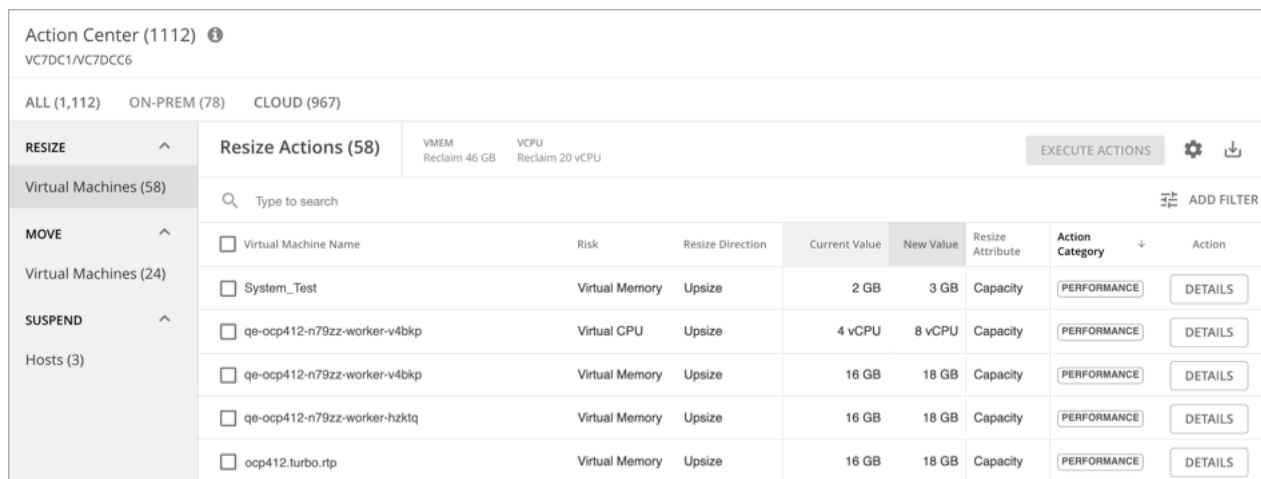
## Utilization Constraints

Utilization constraints affect the actions Turbonomic recommends as it manages your environment. Turbonomic recommends actions that avoid using these resources beyond the given settings. The values you set here specify what percentage of the existing capacity that Turbonomic will consider to be 100% of capacity.

| Attribute             | Default Value |
|-----------------------|---------------|
| Switch Net Throughput | 70            |

## Turbonomic Actions

After you deploy your targets, Turbonomic starts its analysis of your environment. This holistic analysis identifies problems and the actions you can take to *resolve* and *avoid* these problems. Turbonomic then generates a set of actions for that particular analysis and displays it in Action Center for you to investigate.



The screenshot shows the Action Center interface for a target environment (VC7DC1/VC7DCC6). It displays a list of actions under the 'RESIZE' category, specifically 'Resize Actions (58)'. The table lists actions for Virtual Machines, including details like VM Name, Risk, Resize Direction, Current Value, New Value, Resize Attribute, Action Category, and an Action button.

| Virtual Machine Name          | Risk           | Resize Direction | Current Value | New Value | Resize Attribute | Action Category | Action  |
|-------------------------------|----------------|------------------|---------------|-----------|------------------|-----------------|---------|
| System_Test                   | Virtual Memory | Upsize           | 2 GB          | 3 GB      | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-v4b1kp | Virtual CPU    | Upsize           | 4 vCPU        | 8 vCPU    | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-v4b1kp | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-hzktq  | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |
| ocp412.turbo.rtp              | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |

To get the best results from Turbonomic, execute these actions promptly and consider automating as many of them as possible. You can execute these actions from the user interface or outside Turbonomic. To automate these actions, create an [automation policy](#) or change action acceptance to *Automatic* in the default policies.



At first glance, individual actions might appear trivial and it is instinctively convenient to ignore them. It is important to keep in mind that executing a single action can impact other workloads in a meaningful way, helping move these other workloads closer to their desired state. However, if you find that a recommended action is not acceptable (for example, if it violates existing business rules), you can set up a policy with your preferred action.

In some cases, actions can introduce disruptions that you want to avoid at all costs. For example, during critical hours, Turbonomic might execute a resize action on a mission critical resource, which then requires that resource to restart. It is important to anticipate these disruptions and plan accordingly. For example, you can create a group for all critical resources and then schedule the execution of actions to off-peak hours or weekends.

## Working With Action Center

When you start using Turbonomic, all the actions that the product generates appear as pending. Use Action Center to review and execute pending actions.

The screenshot shows the Action Center interface for a cluster named VC7DC1/VC7DCC6. It displays a list of pending actions, specifically 'Resize Actions (58)'. The table below details these actions, showing the virtual machine name, the resource being resized (Virtual Memory or Virtual CPU), the direction of the resize (Upsize), the current and new values, the resize attribute (Capacity), the action category (PERFORMANCE), and an Action button for each entry.

| Virtual Machine Name         | Risk           | Resize Direction | Current Value | New Value | Resize Attribute | Action Category | Action  |
|------------------------------|----------------|------------------|---------------|-----------|------------------|-----------------|---------|
| System_Test                  | Virtual Memory | Upsize           | 2 GB          | 3 GB      | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-v4bkp | Virtual CPU    | Upsize           | 4 vCPU        | 8 vCPU    | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-v4bkp | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |
| qe-ocp412-n79zz-worker-hzktq | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |
| ocp412.turbo.rtp             | Virtual Memory | Upsize           | 16 GB         | 18 GB     | Capacity         | PERFORMANCE     | DETAILS |

## Viewing Action Center

Action Center displays in areas of the product that reference actions, providing a consistent view and experience. Here are some ways you can view Action Center.

- The Action button on the main menu provides easy access to all the current pending actions.

| Volume ID                                            | Tier     | Size    | State      |
|------------------------------------------------------|----------|---------|------------|
| <input type="checkbox"/> test-rosa-28tfg-dynamic-pvc | GP2      | 10 GiB  | Unattached |
| <input type="checkbox"/> ocp47demo-2v5jc-dynamic-    | GP2      | 5 GiB   | Unattached |
| <input type="checkbox"/> ocp47demo-2v5jc-dynamic-    | GP2      | 5 GiB   | Unattached |
| <input type="checkbox"/> mpmmodawsvolumetest01       | Standard | 250 GiB | Unattached |

Action Center in this view is its own dedicated page. Bookmark the page or share it with interested parties.

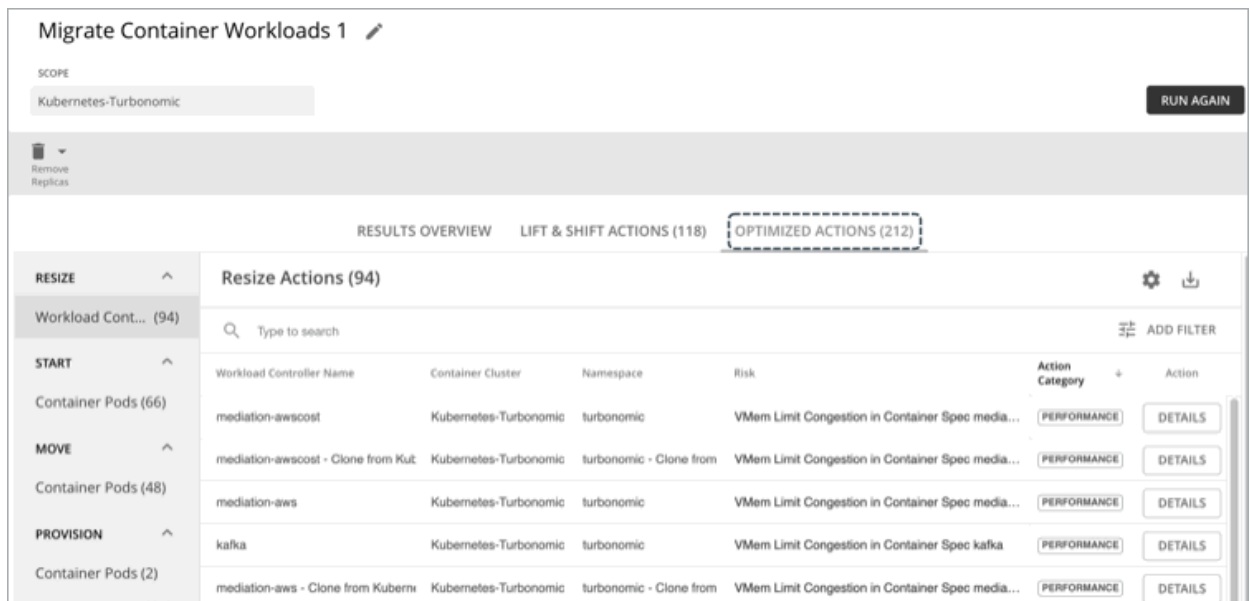
- To focus on actions for a specific entity type, click the entity type in the supply chain, and then click the **Actions** tab.

| Workload Controller Name                   | Container Cluster     | Namespace  |
|--------------------------------------------|-----------------------|------------|
| <input type="checkbox"/> mediation-awscost | Kubernetes-Turbonomic | turbonomic |
| <input type="checkbox"/> mediation-aws     | Kubernetes-Turbonomic | turbonomic |
| <input type="checkbox"/> kafka             | Kubernetes-Turbonomic | turbonomic |
| <input type="checkbox"/> mediation-azure   | Kubernetes-Turbonomic | turbonomic |
| <input type="checkbox"/> mediation-gcpcost | Kubernetes-Turbonomic | turbonomic |

- You can launch Action Center from action-focused charts, such as the Pending Actions, Top Utilized, and Potential Savings charts.

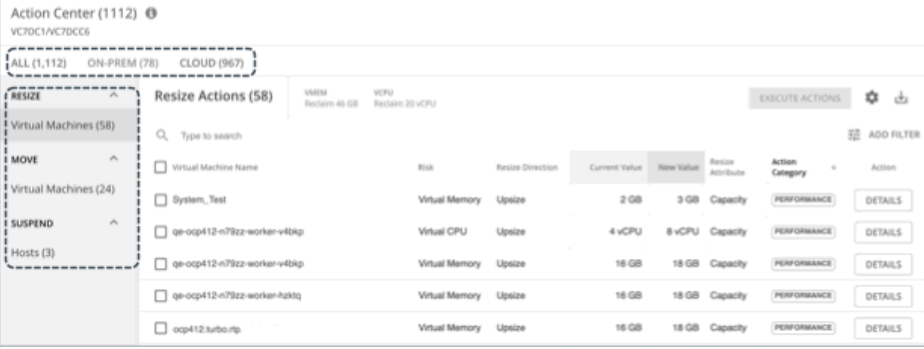

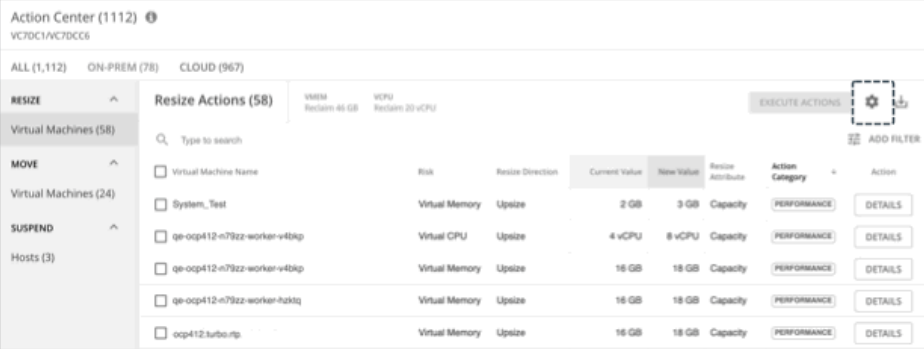


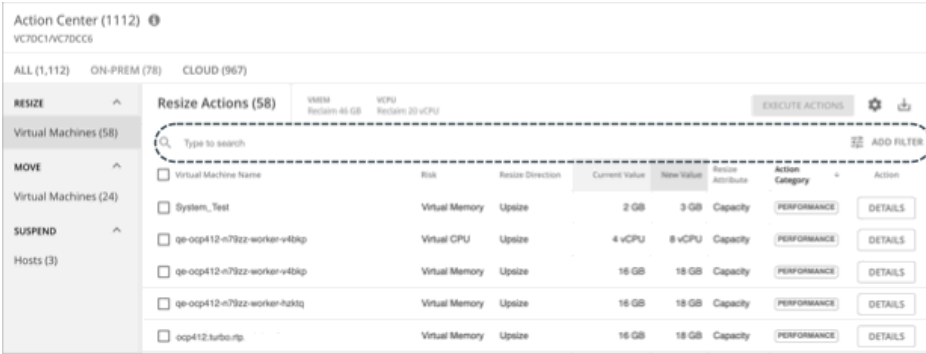
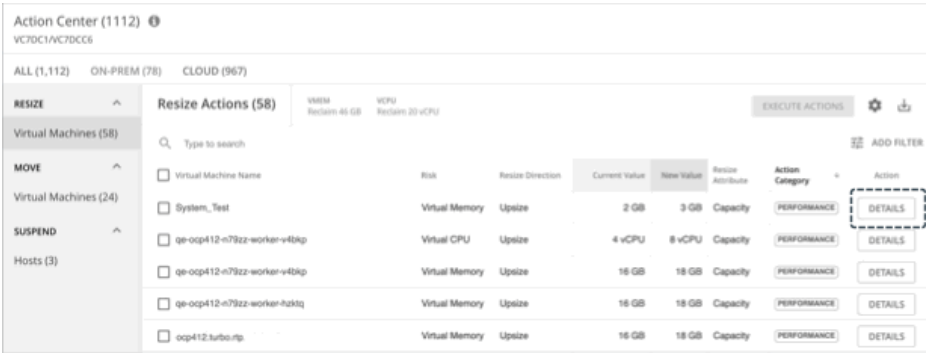
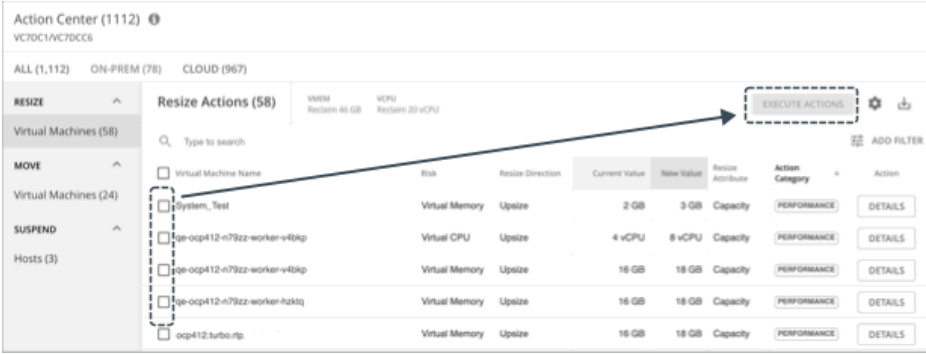
- After you run a plan, click the Actions tab(s) to open Action Center.




## Action Center Features

Action Center provides a comprehensive view of actions. The following features are available to help you work with actions with ease.

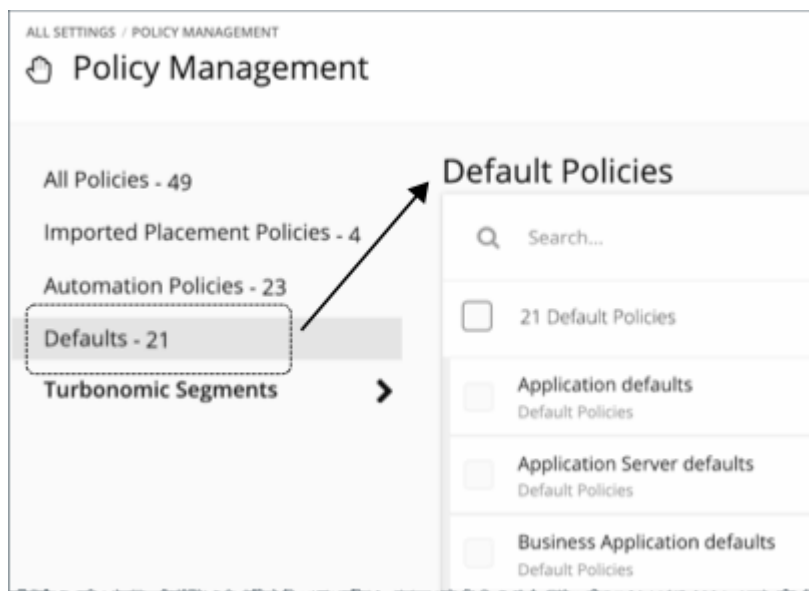
| Feature           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Action groups     | <p>Action Center provides meaningful groups of actions.</p>  <p>The screenshot shows the Action Center interface with a sidebar on the left containing three main action groups: RESIZE (58 actions), MOVE (24 actions), and SUSPEND (3 actions). The RESIZE group is expanded, showing a list of actions for various virtual machines. The table columns include Virtual Machine Name, Risk, Resize Direction, Current Value, New Value, Resize Attribute, Action Category, and Action.</p> <ul style="list-style-type: none"> <li>• Actions are grouped by environment type (on-prem or cloud). For a cloud environment, Action Center also groups actions by cloud provider.</li> <li>• Actions are grouped by <u>action types</u>. These actions may apply to one or several entities.</li> </ul> |
| Table columns     | <p>Actions are presented in tabular format. Columns in the table provide information that is relevant to the action type and entity that you are currently viewing.</p>  <p>The screenshot shows the same Action Center interface as above, but with a dashed box highlighting the table columns: Virtual Machine Name, Risk, Resize Direction, Current Value, New Value, Resize Attribute, Action Category, and Action.</p>                                                                                                                                                                                                                                                                                                                                                                        |
| Configure columns | <p>The Configure columns button allows you to customize the display of columns. You can show or hide certain columns, or rearrange them as needed.</p>  <p>The screenshot shows the same Action Center interface, but with a dashed box highlighting the 'CONFIGURE COLUMNS' button (represented by a gear icon) located in the top right corner of the table area.</p>                                                                                                                                                                                                                                                                                                                                                                                                                             |

| Feature                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Search and Add Filter button</p> | <p>For a long list of actions, use Search or filters to narrow the results. The filter options you see are those that are relevant to the action type and entity that you are currently viewing.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <p>Details button</p>               | <p>The Details button for each action opens the <a href="#">Action Details</a> page, which provides additional information to help you understand why Turbonomic recommends an action and what you would gain if you execute it.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <p>Action execution</p>             | <p>If an action is executable, you can execute it from Action Center. Select one or several actions that you want to execute, and then click <b>Execute Actions</b>.</p>  <ul style="list-style-type: none"> <li>• Some actions may not be executable. For example, the <a href="#">action acceptance</a> setting in a policy may have been set to <i>Recommend</i>, or the underlying technology for the entity does not support automation. The check box for a non-executable action is grayed out. Review the action and then execute it outside Turbonomic.</li> <li>• Some actions can only be executed after certain prerequisites are met. For example, in order to suspend Host A, VM_01 in the host must first</li> </ul> |

| Feature         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <p>move to Host B. However, Host B only has capacity for one VM and is currently hosting VM_02. In this case, Host A suspension is blocked by two prerequisite actions – VM_02 moving to another host and VM_01 moving to Host B.</p> <p>The check box for an action with prerequisites is grayed out and includes the prohibition symbol (⊘). Click the <b>Details</b> button for the action to view prerequisite information. When all the prerequisites have been satisfied, the action becomes executable.</p> |
| Download button | <p>Download the actions you are currently viewing, or all the actions that are currently pending.</p>                                                                                                                                                                                                                                                                                                                            |

## Action Handling

Turbonomic will never execute actions automatically, unless you tell it to. If you examine the default policies that ship with the product, you will notice that these policies do not enable automation on any action. Turbonomic gives you full control over all automation decisions.



At first, your goal may be to evaluate all the actions in Action Center and then execute those that result in immediate improvements in performance and resource utilization. Over time, you develop and fine-tune your action-handling process to meet productivity goals and respond to changing business needs. This process could lead to the following key decisions:

- Disabling actions that should never execute, such as those that violate business rules  
Turbonomic will not consider recommending disabled actions when it performs its analysis.
- Allowing certain actions to execute automatically, such as those that assure performance on mission-critical resources

Automation simplifies your task, while ensuring that workloads continue to have adequate resources to perform optimally. As such, it is important that you set the goal of automating as many actions as possible. This requires evaluating which actions are safe to automate, and on which entities.

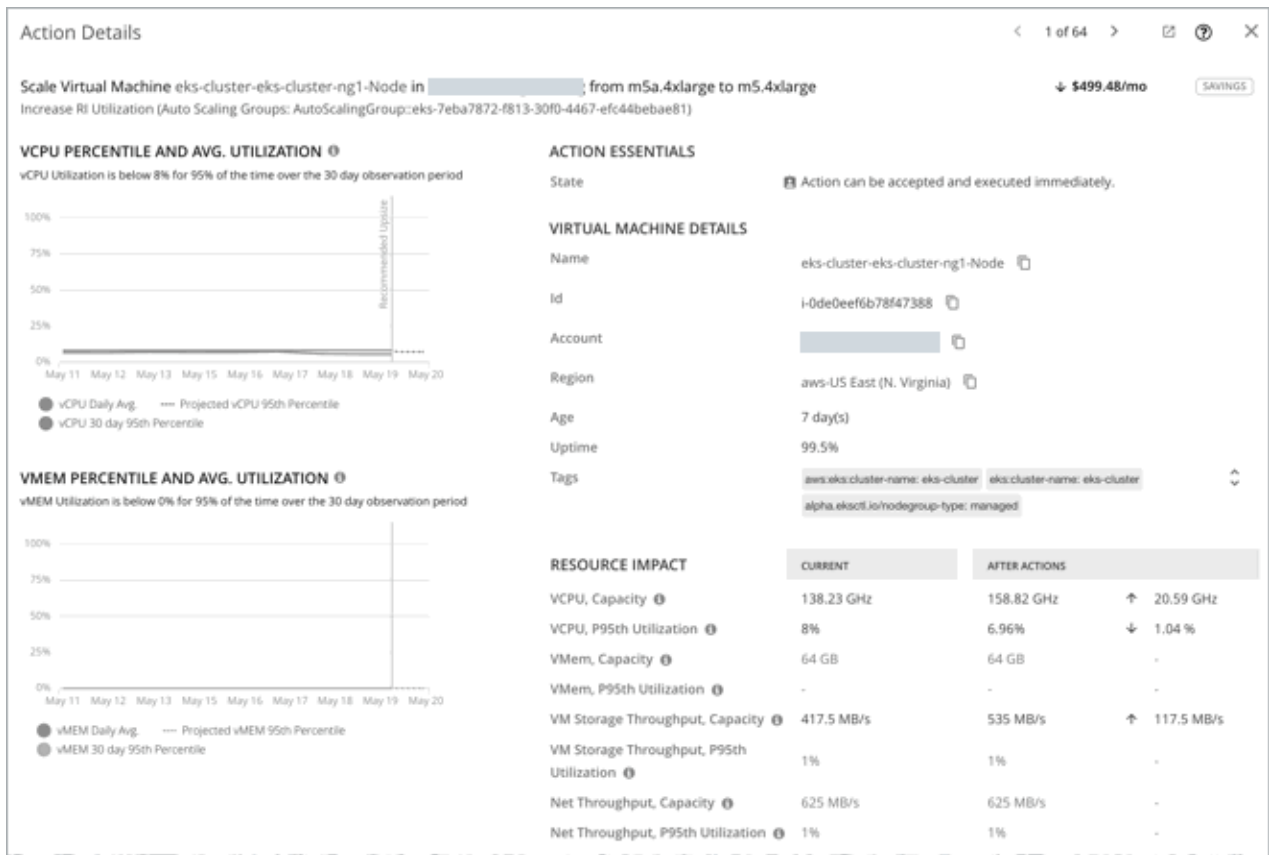
- Continuing to let Turbonomic post certain actions so you can execute them on a case-by-case basis

For example, certain actions might require the approval of specific individuals. In this case, you would want Turbonomic to post those actions for review and only execute the actions that receive an approval.

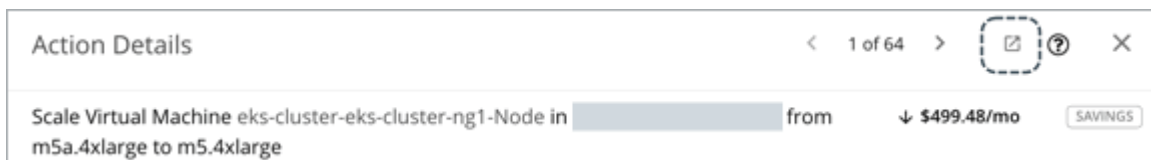
These are the actions that you would look for in Action Center. These actions stop showing after you execute them, if you disable or automate them, or if the environment changes in the next market analysis such that the actions are no longer needed.

## Action Details

The Action Details page provides a comprehensive set of information to help you understand why Turbonomic recommends an action and what you would gain if you execute it.

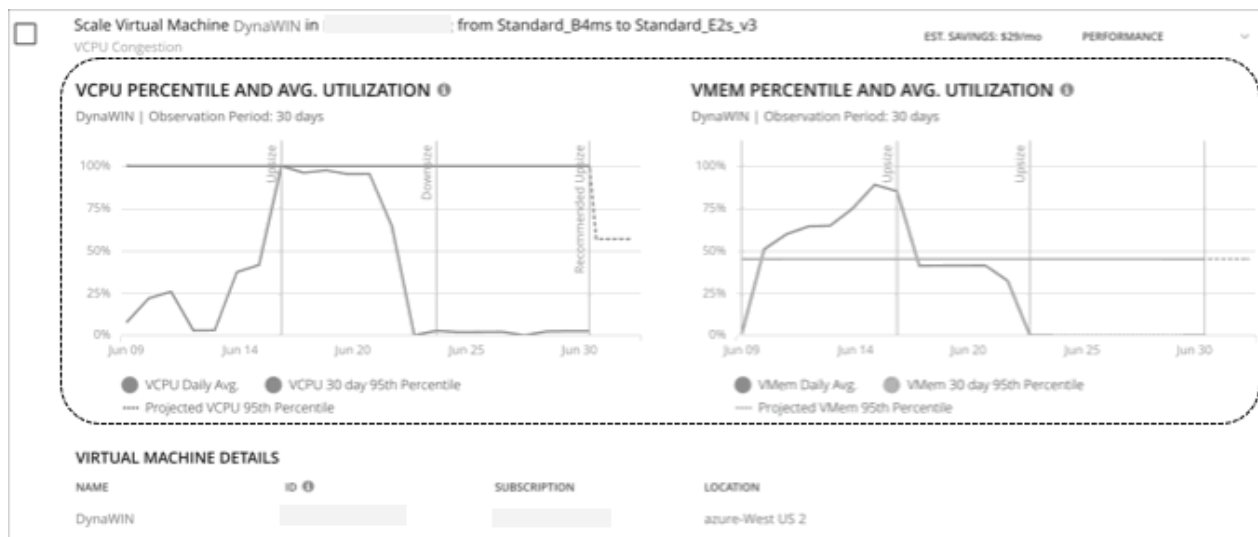


The Action Details page for a specific action has a unique URL (or deep link) that you can share with interested parties. To get the URL, click the 'share' icon at the top-right section of the page and then copy the URL displayed in the address bar of the new tab.



## Utilization Charts

Turbonomic uses percentile calculations to measure resource utilization more accurately, and drive actions that improve overall utilization and reduce costs for cloud workloads. When you examine the details for an entity or pending action, you will see charts that highlight resource *utilization percentiles* for a given observation period, and the projected percentiles after you execute the action.



The charts also plot *daily average utilization* for your reference. If you have previously executed scaling actions on the entity, you can see the resulting improvements in daily average utilization. Put together, these charts allow you to easily recognize utilization trends that drive Turbonomic's recommendations.

### Notes:

- You can set constraints in policies to refine the percentile calculations.
- After you execute an action, it might take some time for the charts to reflect the resulting improvements.

## Entities with Utilization Charts

Utilization charts display for actions on the following entity types:

| Entity Type     | Monitored Resources                                                      |                                                                          | Notes                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|--------------------------------------------------------------------------|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | Percentile Utilization                                                   | Average Utilization                                                      |                                                                                                                                                                                                                                                                                                                                                       |
| Virtual Machine | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> </ul> | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> </ul> | <p>For on-prem VMs, you will see either a VCPU or VMem chart, depending on the commodity that needs to scale. For cloud VMs and VMs in Migrate to Cloud plans, both charts display.</p> <p>These charts also appear when you scope to a given VM (on-prem or cloud) and view the Details page. They also appear in Migrate to Cloud plan results.</p> |



| Entity Type             | Monitored Resources                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                     | Notes                                                                                                                                  |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
|                         | Percentile Utilization                                                                                                                                                                                                                                                        | Average Utilization                                                                                                                                                                                                                                                                                                 |                                                                                                                                        |
| Virtual Machine Spec    | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> </ul>                                                                                                                                                                                                      | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> <li>• Storage</li> <li>• Number of replicas</li> </ul>                                                                                                                                                                                           | See <a href="#">Virtual Machine Spec Actions</a> .                                                                                     |
| Database (cloud)        | <ul style="list-style-type: none"> <li>• DTU Pricing Model <ul style="list-style-type: none"> <li>– DTU</li> </ul> </li> <li>• vCore Pricing Model <ul style="list-style-type: none"> <li>– vCPU</li> <li>– vMem</li> <li>– IOPS</li> <li>– Throughput</li> </ul> </li> </ul> | <ul style="list-style-type: none"> <li>• DTU Pricing Model <ul style="list-style-type: none"> <li>– DTU</li> <li>– Storage</li> </ul> </li> <li>• vCore Pricing Model <ul style="list-style-type: none"> <li>– vCPU</li> <li>– vMem</li> <li>– IOPS</li> <li>– Throughput</li> <li>– Storage</li> </ul> </li> </ul> | See <a href="#">Cloud Database Actions</a> .                                                                                           |
| Database Server (cloud) | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> <li>• IOPS</li> </ul>                                                                                                                                                                                      | <ul style="list-style-type: none"> <li>• vCPU</li> <li>• vMem</li> <li>• IOPS</li> </ul>                                                                                                                                                                                                                            | See <a href="#">Cloud Database Server Actions</a> .                                                                                    |
| Volume (cloud)          | <ul style="list-style-type: none"> <li>• IOPS</li> <li>• Throughput</li> </ul>                                                                                                                                                                                                | <ul style="list-style-type: none"> <li>• IOPS</li> <li>• Throughput</li> </ul>                                                                                                                                                                                                                                      | These charts also appear when you scope to a given volume and view the Details page.<br><br>See <a href="#">Cloud Volume Actions</a> . |
| Workload Controller     | <ul style="list-style-type: none"> <li>• vCPU limits and requests</li> <li>• vMem limits and requests</li> </ul>                                                                                                                                                              | <ul style="list-style-type: none"> <li>• vCPU limits, throttling, and requests</li> <li>• vMem limits and requests</li> </ul>                                                                                                                                                                                       | See <a href="#">Container Actions</a> .                                                                                                |

## Actions by Entity Type

Turbonomic generates actions based on how entity types use or provide resources, and what each entity type supports.

The following tables show the actions that each entity type supports:

## Application Entity Types

| Entity Type          | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Business Application | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a Business Application, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Application lists these actions, thus providing visibility into the risks that have a direct impact on the Business Application's performance.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Business Transaction | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a Business Transaction, but it does recommend actions for the underlying Application Components and infrastructure. The Pending Actions chart for a Business Transaction lists these actions, thus providing visibility into the risks that have a direct impact on the Business Transaction's performance.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Service              | <p><i>For non-Kubernetes Services:</i></p> <p><b>None</b></p> <p>Turbonomic does not recommend actions for non-Kubernetes Services, but it does recommend actions for the underlying Application Components and nodes. The Pending Actions chart for Services list these actions, thus providing visibility into the risks that have a direct impact on their performance.</p> <p><i>For Kubernetes Services:</i></p> <p><b>Provision or Suspend</b></p> <p>For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.</p> <p>For details, see <a href="#">Actions for Kubernetes Services</a>.</p> |

| Entity Type           | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application Component | <p><b>Resize</b></p> <p>Resize the following resources to maintain performance:</p> <ul style="list-style-type: none"> <li>• Thread Pool</li> </ul> <p>Turbonomic generates thread pool resize actions. These actions are recommend-only and can only be executed outside Turbonomic.</p> <ul style="list-style-type: none"> <li>• Connections</li> </ul> <p>Turbonomic uses connection data to generate memory resize actions for on-prem Database Servers.</p> <ul style="list-style-type: none"> <li>• Heap</li> </ul> <p>Turbonomic generates Heap resize actions if an Application Component provides Heap and Remaining GC Capacity, and the underlying VM or container provides VMem. These actions are recommend-only and can only be executed outside Turbonomic.</p> <p><b>Note:</b></p> <p>Remaining GC capacity is the measurement of Application Component uptime that is <i>not</i> spent on garbage collection (GC).</p> <p>The resources that Turbonomic can resize depend on the processes that it discovers from your Applications and Databases targets. Refer to the topic for a specific target to see a list of resources that can be resized.</p> |

## Container Platform Entity Types

| Entity Type    | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service        | <p><b>Provision or Suspend</b></p> <p>For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.</p> <p>For details, see <a href="#">Actions for Kubernetes Services</a>.</p>                                                                                                                         |
| Container      | <p><b>Resize (via Workload Controllers)</b></p> <p>Resize containers to assure optimal utilization of resources. By default, containers resize consistently, which allows all replicas of the same container for the same workload type to resize any resource consistently.</p> <p>Turbonomic shows and executes container resize actions via Workload Controllers. You will <i>not</i> see actions when you set the scope to containers.</p> <p>For details, see <a href="#">Workload Controller Actions</a>.</p> |
| Container Spec | <p><b>None</b></p> <p>A Container Spec retains the historical utilization data of ephemeral containers. Turbonomic uses this data to make accurate container resize decisions, but does not recommend actions for the Container Spec itself.</p>                                                                                                                                                                                                                                                                    |

| Entity Type         | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Namespace           | <p><b>Resize Quota</b></p> <p>Turbonomic treats quotas defined in a namespace as constraints when making container resize decisions. If existing container actions would exceed the namespace quotas, Turbonomic recommends actions to resize up the affected namespace quota.</p> <p>Note that Turbonomic does not recommend actions to resize <i>down</i> a namespace quota. Such an action reduces the capacity that is already allocated to an application. The decision to resize down a namespace quota should include the application owner.</p> <p>For details, see <a href="#">Namespace Actions</a>.</p>                                                                                                                                                                                                                                                                                                                                                       |
| Workload Controller | <p><b>Resize Containers</b></p> <p>A Workload Controller executes <a href="#">container</a> actions. When you set the scope to a Workload Controller and view the actions list, the actions apply to containers. Turbonomic does not recommend actions for the Workload Controller itself.</p> <p>For details, see <a href="#">Workload Controller Actions</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Container Pod       | <ul style="list-style-type: none"> <li>• <b>Move</b></li> </ul> <p>Move a pod between nodes (VMs) to address performance issues or improve infrastructure efficiency. For example, if a particular node is congested for CPU, you can move pods to a node with sufficient capacity. If a node is underutilized and is a candidate for suspension, you must first move the pods before you can safely suspend the node.</p> <ul style="list-style-type: none"> <li>• <b>Provision/Suspend</b></li> </ul> <p>For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, provision or suspend pods associated with those Services to maintain SLOs for your applications.</p> <p>When recommending node provision or suspend actions, Turbonomic will also recommend provisioning pods (based on demand from DaemonSets) or suspending the related pods.</p> <p>For details, see <a href="#">Container Pod Actions</a>.</p> |
| Container Cluster   | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a Container Cluster. Instead, it recommends actions for the containers, pods, nodes (VMs), and volumes in the cluster. Turbonomic shows all of these actions when you scope to a Container Cluster and view the Pending Actions chart.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Entity Type          | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kubernetes node (VM) | <p>A Kubernetes node (cloud or on-prem) is represented as a Virtual Machine entity in the supply chain.</p> <ul style="list-style-type: none"> <li>• <b>Provision</b><br/>Provision nodes to address workload congestion or meet application demand.</li> <li>• <b>Suspend</b><br/>Suspend nodes after you have consolidated pods or defragmented node resources to improve infrastructure efficiency.</li> <li>• <b>Reconfigure</b><br/>Reconfigure nodes that are currently in the NotReady state.</li> </ul> <p><b>Note:</b><br/>For nodes in the public cloud, Turbonomic reports the cost savings or investments attached to these actions.<br/>For details, see <a href="#">Node Actions</a>.</p> |

### Cloud Infrastructure Entity Types

| Entity Type             | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual Machine (Cloud) | <p>For details about cloud VM actions, see the following topics:</p> <ul style="list-style-type: none"> <li>• <a href="#">Actions for AWS VMs</a></li> <li>• <a href="#">Actions for Azure VMs</a></li> <li>• <a href="#">Actions for GCP VMs</a></li> </ul>                                                                                                                                                                |
| Virtual Machine Spec    | <ul style="list-style-type: none"> <li>• <b>Scale</b><br/>Scale Azure App Service plans to optimize app performance or reduce costs, while complying with business policies.</li> <li>• <b>Delete</b><br/>Delete empty Azure App Service plans as a cost-saving measure. A plan is considered empty if it is not hosting any running apps.</li> </ul> <p>For details, see <a href="#">Virtual Machine Spec Actions</a>.</p> |
| App Component Spec      | <p><b>None</b></p> <p>Turbonomic does not recommend actions for App Component Specs, but it does recommend actions for the underlying Virtual Machine Specs. For details, see <a href="#">Virtual Machine Spec Actions</a>.</p>                                                                                                                                                                                             |

| Entity Type             | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database (Cloud)        | <p><b>Scale (SQL Database)</b></p> <ul style="list-style-type: none"> <li>DTU Pricing Model<br/>Scale DTU and storage resources to optimize performance and costs.</li> <li>vCore Pricing Model<br/>Scale vCPU, vMem, IOPS, throughput and storage resources to optimize performance and costs.</li> </ul> <p><b>Suspend (Dedicated SQL Pool)</b></p> <p>Suspend an idle dedicated SQL pool (used for Azure Synapse Analytics) to reduce compute costs.</p> <p><b>Note:</b></p> <p>Currently, Turbonomic does not generate actions to start a suspended pool. You can manually start a pool from Azure.</p> <p>For details, see <a href="#">Cloud Database Actions</a>.</p> |
| Database Server (Cloud) | <ul style="list-style-type: none"> <li><b>Stop</b> and <b>Start</b> (also known as 'parking' actions)<br/>Stop a Database Server for a given period of time to reduce your cloud expenses, and then start it at a later time.</li> <li><b>Scale</b><br/>Scale compute and storage resources to optimize performance and costs.</li> </ul> <p>For details, see <a href="#">Cloud Database Server Actions</a> and <a href="#">Parking: Stop or Start Cloud Resources</a>.</p>                                                                                                                                                                                                 |
| Volume (Cloud)          | <ul style="list-style-type: none"> <li><b>Scale</b><br/>Scale attached volumes to optimize performance and costs.</li> <li><b>Delete</b><br/>Delete unattached volumes as a cost-saving measure. Turbonomic generates an action immediately after discovering an unattached volume.</li> </ul> <p>For details, see <a href="#">Cloud Volume Actions</a>.</p>                                                                                                                                                                                                                                                                                                                |
| Zone                    | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a cloud zone.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Region                  | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a cloud region.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

## On-prem Infrastructure Entity Types

| Entity Type               | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual Machine (On-prem) | <ul style="list-style-type: none"> <li data-bbox="605 264 716 289">• <b>Resize</b></li> <li data-bbox="630 310 1472 562">When multiple resize actions that relate to the same vCenter VM are accepted (manually or automated), they execute together to avoid multiple restarts for the same VM. For example, a VM has both a vCPU resize action and a vMEM resize action. When selecting both actions and clicking Execute in the UI, the actions combine into the fewest vCenter tasks to minimize the disruption on the VM instead of one set of tasks per action. Only the VMware vCenter integration supports this functionality at this time.</li> <li data-bbox="630 583 954 609">– Resize resource capacity</li> <li data-bbox="654 630 1472 850">Change the capacity of a resource that is allocated for the VM. For example, a resize action might recommend increasing the VMem available to a VM. Before recommending this action, Turbonomic verifies that the VM's cluster can adequately support the new size. If the cluster is highly utilized, Turbonomic will recommend a move action, taking into consideration the capacity of the new cluster and compliance with existing placement policies.</li> <li data-bbox="654 871 1472 955">For hypervisor targets, Turbonomic can resize vCPU by changing the VM's socket or cores per socket count. For details, see <a href="#">vCPU Scaling Controls</a>.</li> <li data-bbox="630 976 990 1001">– Resize resource reservation</li> <li data-bbox="654 1022 1472 1180">Change the amount of a resource that is reserved for a VM. For example, a VM could have an excess amount of memory reserved. That can cause memory congestion on the host — A resize action might recommend reducing the amount reserved, freeing up that resource and reducing congestion</li> <li data-bbox="630 1201 909 1226">– Resize resource limit</li> <li data-bbox="654 1247 1472 1362">Change the limit that is set on the VM for a resource. For example, a VM could have a memory limit set on it. If the VM is experiencing memory shortage, an action that decreases or removes the limit could improve performance on that VM.</li> <li data-bbox="605 1383 1101 1409">• <b>Resize PU / Resize VP (IBM PowerVM)</b></li> <li data-bbox="630 1430 1472 1545">Turbonomic does not support native execution of Resize PU (processing units) and Resize VP (virtual processor) actions. However, actions can be executed using external automation platforms through Action Orchestration workflows such as Action Scripts or Webhooks.</li> <li data-bbox="630 1566 998 1591">– Min/Max Observation Period</li> <li data-bbox="654 1612 1372 1638">Observation period settings only apply to Resize VP actions.</li> <li data-bbox="654 1659 1339 1684">For details, see <a href="#">Aggressiveness and Observation Periods</a>.</li> <li data-bbox="630 1705 1128 1730">– Increment Constant for Processing Unit</li> <li data-bbox="654 1751 1388 1814">Change the increment constant value to use for VM PU resize actions. PUs can be a decimal value in a VM setting.</li> <li data-bbox="654 1835 1323 1860">For details, see <a href="#">Increment Constant for Processing Unit</a>.</li> <li data-bbox="630 1881 1161 1906">– Increment Constant for Virtual Processors</li> <li data-bbox="654 1927 1388 1990">Change the increment constant value to use for VM VP resize actions. VPs must be an integer value in a VM setting.</li> <li data-bbox="654 2011 1356 2037">For details, see <a href="#">Increment Constant for Virtual Processors</a>.</li> <li data-bbox="605 2058 698 2083">• <b>Move</b></li> </ul> |

| Entity Type               | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Volume (On-prem)          | <ul style="list-style-type: none"> <li>• <b>Move</b><br/>Move a VM's volume (virtual storage) due to excess utilization of the current datastore, or for more efficient utilization of datastores in the environment.<br/><br/>Points to consider: <ul style="list-style-type: none"> <li>– The default global policy includes a setting that directs Turbonomic to use relevant metrics when analyzing and recommending actions for volumes. For details, see <a href="#">Enable Analysis of On-prem Volumes</a>.</li> <li>– Turbonomic will not recommend moving a volume to a datastore that is currently in maintenance mode. Any volume in that datastore should move to an active datastore (for example, via vMotion).</li> </ul> </li> <li>• <b>Reconfigure</b><br/>Reconfigure a VM's volume (virtual storage) to comply with placement policies.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Database Server (On-prem) | <p><b>Resize</b></p> <p>Resize the following resources:</p> <ul style="list-style-type: none"> <li>• Connections<br/>Turbonomic uses connection data to generate memory resize actions for on-prem Database Servers.</li> <li>• Database memory (DBMem)<br/><br/>Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.<br/><br/>A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.<br/><br/>When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.</li> <li>• Transaction log<br/><br/>Resize actions based on the transaction log resource depend on support for virtual storage in the underlying hypervisor technology.<br/><br/>Currently, Turbonomic does not support resize actions for Oracle and Database Servers on the Hyper-V platform (due to the lack of API support for virtual storage).</li> </ul> |
| Virtual Data Center       | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a Virtual Data Center. Instead, it recommends actions for the entities that provide resources to the Virtual Data Center.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



| Entity Type   | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Business User | <p><b>Move</b></p> <p>Move a Business User between desktop pools to address:</p> <ul style="list-style-type: none"> <li>• Resource congestion on the image</li> </ul> <p>When utilization is consistently near capacity for image resources, Turbonomic can recommend moving a Business User to a desktop pool that serves larger images.</p> <ul style="list-style-type: none"> <li>• Resource congestion on the desktop pool</li> </ul> <p>When utilization is consistently near capacity for the desktop pool, Turbonomic can recommend moving a Business User to a desktop pool that has more available resources.</p> <p><b>Note:</b></p> <p>To support moves, you must configure placement policies that merge <i>similarly configured</i> desktop pools. For details, see <a href="#">Desktop Pool Placement Policies</a>.</p>                          |
| Desktop Pool  | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a desktop pool. It recommends actions for the Business Users running active sessions in the pool.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| View Pod      | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a view pod. Instead, it recommends actions for the Business Users that are running active sessions.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Host          | <ul style="list-style-type: none"> <li>• <b>Start</b></li> </ul> <p>Start a suspended host when there is increased demand for physical resources.</p> <ul style="list-style-type: none"> <li>• <b>Provision</b></li> </ul> <p>Provision a new host in the environment when there is increased demand for physical resources. Turbonomic can then move workloads to that host.</p> <ul style="list-style-type: none"> <li>• <b>Suspend</b></li> </ul> <p>When physical resources are underutilized on a host, move existing workloads to other hosts and then suspend the host.</p> <ul style="list-style-type: none"> <li>• <b>Reconfigure</b></li> </ul> <p>Turbonomic generates this action in response to changing demand for software licenses. For details, see <a href="#">License Policy</a>.</p> <p>For details, see <a href="#">Host Actions</a>.</p> |
| Chassis       | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a chassis.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Entity Type  | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Center  | <p><b>None</b></p> <p>Turbonomic does not recommend actions for a data center. Instead, it recommends actions for the entities running in the data center.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Storage      | <ul style="list-style-type: none"> <li data-bbox="609 359 699 386">• <b>Move</b><br/>For high utilization of physical storage, move datastore to a different disk array (aggregate).</li> <li data-bbox="609 478 748 506">• <b>Provision</b><br/>For high utilization of storage resources, provision a new datastore.</li> <li data-bbox="609 569 716 596">• <b>Resize</b><br/>Increase or decrease the datastore capacity.</li> <li data-bbox="609 659 699 686">• <b>Start</b><br/>For high utilization of storage resources, start a suspended datastore.</li> <li data-bbox="609 749 742 777">• <b>Suspend</b><br/>For low utilization of storage resources, move served VMs to other datastores and suspend this one.</li> <li data-bbox="609 869 716 896">• <b>Delete</b><br/>Delete a datastore or volume that has been suspended for a period of time.</li> </ul> <p>For details, see <a href="#">Storage Actions</a>.</p> |
| Logical Pool | <ul style="list-style-type: none"> <li data-bbox="609 1066 711 1094">• Resize</li> <li data-bbox="609 1108 743 1136">• Provision</li> <li data-bbox="609 1150 699 1178">• Move</li> <li data-bbox="609 1192 695 1220">• Start</li> <li data-bbox="609 1234 737 1262">• Suspend</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| Entity Type        | Supported Actions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Disk Array         | <ul style="list-style-type: none"> <li data-bbox="610 212 1464 321">• <b>Provision</b><br/>For high utilization of the disk array's storage, provision a new disk array (recommendation, only).</li> <li data-bbox="610 331 1464 441">• <b>Start</b><br/>For high utilization of disk array, start a suspended disk array (recommendation, only).</li> <li data-bbox="610 451 1464 594">• <b>Suspend</b><br/>For low utilization of the disk array's storage, move VMs to other datastores and suspend volumes on the disk array (recommendation, only).</li> <li data-bbox="610 604 1464 888">• <b>Move</b><br/>(Only for NetApp Cluster-Mode) For high utilization of Storage Controller resources, Turbonomic can move an aggregate to another storage controller. The storage controllers must be running.<br/><br/>For high IOPS or latency, a move is always off of the current disk array. All the volumes on a given disk array show the same IOPS and Latency, so moving to a volume on the same array would not fix these issues.</li> <li data-bbox="610 898 1464 1182">• <b>Move VM</b><br/><br/>For high utilization of Storage on a volume, Turbonomic can move a VM to another volume. The new volume can be on the current disk array, on some other disk array, or on any other datastore.<br/><br/>For high IOPS or latency, a move is always off of the current disk array. All the volumes on a given disk array show the same IOPS and Latency, so moving to a volume on the same array would not fix these issues.</li> <li data-bbox="610 1192 1464 1302">• <b>Move Datastore</b><br/><br/>To balance utilization of disk array resources, Turbonomic can move a datastore to another array.</li> </ul> |
| Storage Controller | <p data-bbox="610 1350 1464 1459"><b>Provision</b><br/>For high utilization of the storage controller's CPU, provision a new storage controller, and then move disk arrays to it.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| IO Module          | <p data-bbox="610 1493 1464 1566"><b>None</b><br/>Turbonomic does not recommend actions for an IO Module.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Switch             | <p data-bbox="610 1608 1464 1682"><b>Resize</b><br/>Resize PortChannel for a switch to increase bandwidth.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## Action Categories

Turbonomic groups entries in the Actions List by different categories. These categories do not strictly define the severity of an issue, but they indicate the nature of the issue.

## Compliance

A virtual environment can include policies that limit availability of resources. It's possible that the environment configuration violates these defined policies. In such cases, Turbonomic identifies the violation and recommends actions that bring the entity back into compliance.

## Efficiency

Efficient utilization of resources is an important part of running in the desired state. Running efficiently maximizes your investment and reduces cost. When Turbonomic discovers underutilized resources, it recommends actions to consolidate your operations. For example, it can recommend that you move certain VMs onto a different host, and then suspend the original host.

## Performance

Ultimately, the reason to manage workloads in your environment is to assure performance and meet productivity goals. Turbonomic can recommend actions when it detects conditions that directly put performance at risk. You can consider these critical conditions, and you should execute the recommended actions as soon as possible.

## Prevention

Turbonomic constantly monitors conditions, and works to keep your environment running in a desired state. As it finds issues that risk moving the environment out of this state, it recommends preventive actions. You should attend to these issues, and perform the associated actions. If you do not, the environment may drift away from the desired state, and performance may be put at risk.

## Action Types

Turbonomic performs the following general types of actions:

- Buy – Purchase cloud instances at a discounted rate to reduce on-demand costs
- Delete – Remove unused entities (for example, datastores on disk arrays or unattached volumes)
- Move – Move an entity from one provider to another to address performance issues or improve infrastructure efficiency
- Optimization – Increase discount coverage for cloud workloads to reduce on-demand costs
- Provision – Add resource providers to the environment to increase capacity
- Reconfigure – Correct a misconfiguration to bring an entity into compliance
- Scale/Resize – Change the resources allocated to workloads to meet demand
- Start – Start a stopped/suspended entity
- Stop/Suspend – Pause an entity for a period of time to increase efficient use of resources

For information on the actions that Turbonomic generates for specific entities, see [Actions by Entity Type](#).

## Placement

Placement actions determine the best provider for a consumer. These include initial placement for a new entity, and move actions that change a consumer to use a different provider. For example, moving a VM assigns it to a different host. Moving a VM's storage means the VM will use a different datastore.

## Placement Constraints

When making placement decisions, Turbonomic checks for placement constraints to limit the set of providers for a given consumer. It respects automatic placement constraints, including cluster boundaries and DRS rules. It also considers user-configured constraints defined in a placement policy to ensure compliance to specific business requirements.

Reviewing the constraints on an entity helps you understand the actions that Turbonomic recommends. If an action seems questionable to you, then you should look at the constraints on the affected entities. It's possible that some policy or constraint is in effect, and it keeps Turbonomic from recommending a more obvious action. For details, see [Entity Placement Constraints](#).

You can run plans to see what happens if you turn off constraints, or disable or enable certain placement policies.

## Effective CPU Capacity

CPU processor speed is not necessarily an effective indicator of CPU capacity. For example, processor architecture can make a slower CPU have a greater effective capacity. Newer models of machines can often have fewer cores or less clock speed, but still have a higher effective capacity.

When placing VMs on hosts in the on-prem environment, Turbonomic discovers the effective CPU capacity of your hosts. This increases the accuracy of placement calculations so that newer, more efficient hosts will show a greater effective capacity than less efficient hosts that might have larger or faster processors.

To discover the effective capacity, Turbonomic uses benchmark data from [spec.org](#). This benchmark data maps to effective capacity settings that Turbonomic uses to make placement calculations.

You can see a catalog of these benchmark data and choose from listed processors when you edit Host templates. For more information, see [Selecting CPUs from the Catalog](#).

## Shared-Nothing Migration Actions

If you have enabled both storage and VM moves, Turbonomic can perform shared-nothing migrations, which move the VM and the stored VM files simultaneously. For details, see [Shared-Nothing Migration](#).

## Cross-vCenter vMotion

VMware vSphere 6.0 introduces functionality that enables migration of virtual machines between different vCenter Server instances. Turbonomic supports this capability through *Merge* placement policies (see [Creating Placement Policies](#)). It considers cross-vCenter locations when calculating placement, and can recommend or execute moves to different vCenter servers.

## Moves on the Public Cloud

On the public cloud you do not place workloads on physical hosts. In the Turbonomic Supply Chain, the Host nodes represent availability zones. Turbonomic can recommend moving a workload to a different zone, if such a move can reduce your cloud cost. These moves recognize constraints, such as availability of instances types and [discounts](#) in the given zones.

In AWS environments, a VM can use Elastic Block Stores (EBS) or Instance Storage. If the VM's root storage is EBS, then Turbonomic can recommend a VM move. However, because Instance Storage is ephemeral and a move would lose the stored data, Turbonomic does not recommend moving a VM that has Instance Storage as its root storage.

If a VM is running within a billing family, then Turbonomic only recommends moving that VM to other regions within that billing family.

In AWS environments that use RIs, Turbonomic recognizes Availability Zones that you have specified for your RI purchases. For move and resize actions, Turbonomic gives precedence to these RIs in the given zone. All else being equal for a given zone, if you have Zone RIs with reserved capacity and RIs that do not reserve capacity, Turbonomic will use the Zone RI first.

## Scaling

Scaling actions update capacity in your environment. For vertical scaling, Turbonomic increases or decreases the capacity of resources on existing entities. For horizontal scaling it provisions new providers.

For example, provisioning a host adds more compute capacity that is available to run VMs. Provisioning a VM adds capacity to run applications.

Turbonomic can provision the following:

- Containers
- VMs
- Hosts
- Storage
- Storage Controllers (only for planning scenarios)
- Disk Arrays

Under certain circumstances, Turbonomic can also recommend that you provision a virtual datacenter.

## Discount Optimization

To reduce your cloud costs, Turbonomic can recommend scaling VMs to instance types that are charged discounted rates.

### Note:

Under most circumstances, when a cloud provider offers a new instance type that is meant to replace an older type, the provider offers it at a lower cost. However, a provider may provide a new instance type with identical costs as the older instance types. If this occurs, and capacity and cost are equal, Turbonomic cannot ensure that it chooses the newer instance type. To work around this issue, you can create an Action Automation policy that excludes the older instance type.

- Discount Utilization

This chart shows how well you have utilized your current discount inventory. The desired goal is to maximize the utilization of your inventory and thus take full advantage of the discounted pricing offered by your cloud provider.

- Discount Coverage

This chart shows the percentage of VMs covered by discounts. If you have a high percentage of on-demand VMs, you should be able to reduce your monthly costs by increasing coverage. To increase coverage, you scale VMs to instance types that have existing capacity.

- Discount Inventory

This chart lists the cloud provider discounts discovered in your environment.

Discount optimization actions are not executed by Turbonomic users. They reflect capacity reassignments performed by your cloud provider.

## Configuration

These are reconfigure and resize actions. Reconfigure actions can add necessary network access, or reconfigure storage. Resize actions allocate more or less resource capacity on an entity, which can include adding or reducing VCPUs or VMem on a VM, adding or reducing capacity on a datastore, and adding or reducing volumes in a disk array.

Turbonomic can reconfigure the following:

- VMs
- Containers
- Storage
- Disk Arrays
- Virtual Datacenters

## Start/Buy

Turbonomic can recommend that you start a suspended entity to add capacity to the environment. It can also recommend purchasing cloud provider [discounts](#) to reduce costs for your current workload.

## Stop

Stop actions suspend entities without removing them from the environment. Suspended capacity is still available to be brought back online, but is currently not available for use. Suspended resources are candidates for termination.

Turbonomic can suspend the following:

- Application Components
- Container Pods
- Disk Arrays
- Hosts
- Storage (on-prem)
- Virtual datacenter

## Delete

Delete actions affect storage. For example, Turbonomic might recommend that you delete wasted files to free up storage space, or delete unused storage in your cloud environment to reduce storage costs.

## Wasted Storage in Azure Environments

In Azure environments, Turbonomic can identify unmanaged storage as unattached volumes, recommend that you remove this unused storage, and then show *estimated* savings after you remove this storage and no longer pay for it. The savings that Turbonomic shows are estimates based on the overall cost for that storage, since Azure does not provide specific values for the cost per volume or cost for the amount of storage that is in use for a given volume. If the estimated savings appear unusually high, then you should identify which storage the actions will remove, and review your billing to calculate the costs with more precision.

## Action Acceptance Modes

Action acceptance modes specify the degree of automation for the generated actions. For example, in some environments you might not want to automate resize down of VMs because that is a disruptive action. You would use action acceptance modes in a policy to set that business rule.

Turbonomic supports the following action acceptance modes:

- Recommend — Recommend the action so a user can execute it via the given hypervisor or by other means
- Manual — Recommend the action, and provide the option to execute that action through the Turbonomic user interface
- Automatic — Execute the action automatically

For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to execute all actions at once. Any action awaiting execution stays in queue. For example, if a VM has both vCPU and vMem resize actions, Turbonomic could resize vCPU first. After this resize completes, it waits five minutes before resizing vMem.

The Pending Actions charts only count actions in *Recommend* or *Manual* mode.

Automated actions appear in the following charts:

- **All Actions** chart on the **Home Page** and the On-prem Executive Dashboard
- **Accepted Actions** chart on the **Home Page**

## Setting Action Acceptance Modes

There are two ways to configure action acceptance modes:

- Change the action acceptance mode in a [default policy](#).
- Create an [automation policy](#), scope the policy to specific entities or groups, and then select the action acceptance mode for each action.

Turbonomic allows you to create dynamic groups to ensure that entities discovered in the future automatically add to a group and apply the policy of that group. If a conflict arises as a result of an entity belonging to several groups, the entity applies the policy with the most conservative action.

## Action Acceptance Overrides

Under some conditions, Turbonomic changes the action acceptance mode of an action from *Manual* to *Recommend*.

Turbonomic makes this change as a safeguard against executing actions that the underlying infrastructure cannot support. For example, assume you have VM move actions set to *Manual*. Then assume Turbonomic analysis wants to move a VM onto a host that is already utilized fully. In this case, there would be other actions to move workloads *off* of the given host to make room for this new VM. However, because moves are *Manual*, the host might not be properly cleared off yet. In that case, Turbonomic changes actions to move workloads *to* the host from *Manual* to *Recommend*.

For cloud environments, some instances require workloads to be configured in specific ways before they can move to those instance types. If Turbonomic recommends moving a workload that is not suitably configured onto one of these instances, then it changes the action acceptance mode from *Manual* to *Recommend*, and then describes the reason.

## Plans: Looking to the Future

The screenshot displays the Turbonomic Plan Page for a workload named 'Add Workload 14'. The page is divided into a configuration sidebar and a main results overview section.

**Configuration:**

- SCOPE: DC14\DC14-Cluster
- Buttons: Add, Replace, Remove, Actions, Ignore Constraints, Placement Policies, Utilization, Baseline, Desired State, Projection
- Product Trust: 1
- Max Host Utilization Level: 80%
- Hosts Action Settings: Provision for Hosts enabled, Suspend for Hosts enabled
- Virtual Machines Action Settings: Scale for Virtual Machines enabled

**RESULTS OVERVIEW PLAN ACTIONS (170)**

Plan has 38 unplaced workloads

**Plan Summary**

|                  | Current | After Plan | Difference | %          |
|------------------|---------|------------|------------|------------|
| Virtual Machines | 30      | 30         | 0          | 0 %        |
| Hosts            | 3       | 15         | 12         | ▲ 400 %    |
| Storage          | 6       | 14         | 8          | ▲ 133.3 %  |
| CPU              | 6 Cores | 32 Cores   | 26         | ▲ 433.3 %  |
| Memory           | 12 GB   | 246.5 GB   | 234.5 GB   | ▲ 1958.3 % |
| Storage Amount   | 3.6 GB  | 6.3 GB     | 2.7 GB     | ▲ 50 %     |
| Host Density     | 10:1    | 2:1        | 8:1        | ▼ 80 %     |
| Storage Density  | 5:1     | 2:1        | 3:1        | ▼ 60 %     |

Show all

Use the Plan Page to run simulations for what-if scenarios that explore possibilities such as:

- Reducing cost while assuring performance for your workloads



- Impact of scaling resources
- Changing hardware supply
- Projected infrastructure requirements
- Optimal workload distribution to meet historical peaks demands
- Optimal workload distribution across existing resources

## How Plans Work

To run a plan scenario, Turbonomic creates a snapshot copy of your real-time market and modifies that snapshot according to the scenario. It then uses the Economic Scheduling Engine to perform analysis on that plan market. A scenario can modify the snapshot market by changing the workload, adding or removing hardware resources, or eliminating constraints such as cluster boundaries or placement policies.

As it runs a plan, Turbonomic continuously analyzes the plan market until it arrives at the optimal conditions that market can achieve. When it reaches that point, the Economic Scheduling Engine cannot find better prices for any of the resources demanded by the workload — the plan stops running, and it displays the results as the plan's desired state. The display includes the resulting workload distribution across hosts and datastores, as well as a list of actions the plan executed to achieve the desired result.

For example, assume a scenario that adds virtual machines to a cluster. To run the plan, Turbonomic takes a snapshot of the current market, and adds the VMs to the specified cluster. Turbonomic then runs analysis on the plan market, where each entity in the supply chain shops for the resources it needs, always looking for a better price — looking for those resources from less-utilized suppliers. This analysis continues until all the resources are provided at the best possible price.

The results might show that you can add more workload to your environment, even if you reduce compute resources by suspending physical machines. The recommended actions would then indicate which hosts you can take offline, and how to distribute your virtual machines among the remaining hosts.

## Idle Workloads

Plans calculate optimal placement and optimal resource allocation for the given workload. However, plans do not include *idle* workloads. This is because an idle VM shows no utilization, so the plan cannot determine optimal placement or what percentage of allocated resources that workload will require when it restarts.

## Plan Management

The screenshot shows the 'Plan Management' interface. On the left is a dark sidebar with navigation icons: 'on' (home), 'SEARCH', 'PLAN', 'PLACE', 'DASHBOARD', and 'SETTINGS'. The main header area contains 'Plan Management' with a triangle icon, and two tabs: 'USER PLAN' (selected) and 'NIGHTLY PLAN CONFIGURATION'. A 'NEW PLAN' button is in the top right. Below the tabs is a list of plans. At the top of the list is a search bar and a 'FILTER' icon. The list shows 23 plans in total. Three items are visible: 'Add Workload 1', 'Add Workload 1', and 'Custom 11'. Each item has a checkbox on the left and a 'RAN AT 09 JAN 2020 04:4' timestamp on the right.

The Plan Management Page is your starting point for creating new plans, viewing saved plans, and deleting saved plans that you don't need anymore. To display this page, click **Plan** in the Turbonomic navigation bar.

- Create new plans

To create a new plan, click the **NEW PLAN** button. See [Setting Up Plan Scenarios](#).

- View saved plans

After you create and run a plan, Turbonomic saves it and then shows it in the Plan Management Page. You can open the saved plan to review the results, or you can change its configuration and run it again.

**Note:**

You can also view saved plans from the Search page, under the **Plans** category.

- Delete saved plans

To delete a saved plan, turn on the plan's checkbox and then click the **Delete** button.

- Configure nightly plans

Turbonomic runs nightly plans to calculate headroom for the clusters in your on-prem environment. For each cluster plan, you can set which VM template to use in these calculations. See [Configuring Nightly Plans](#).

## Setting Up Plan Scenarios

A plan scenario specifies the overall configuration of a plan. Creating the plan scenario is how you set up a what-if scenario to see the results you would get if you changed your environment in some way.

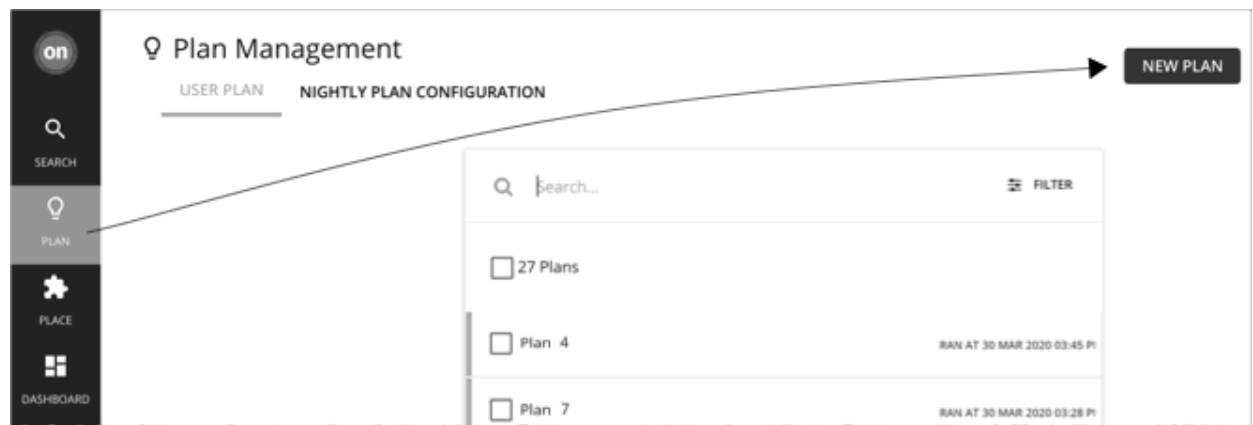
This topic walks you through the general process of setting up a plan scenario.

### 1. Plan Entry Points

You can begin creating a plan scenario from different places in the user interface:

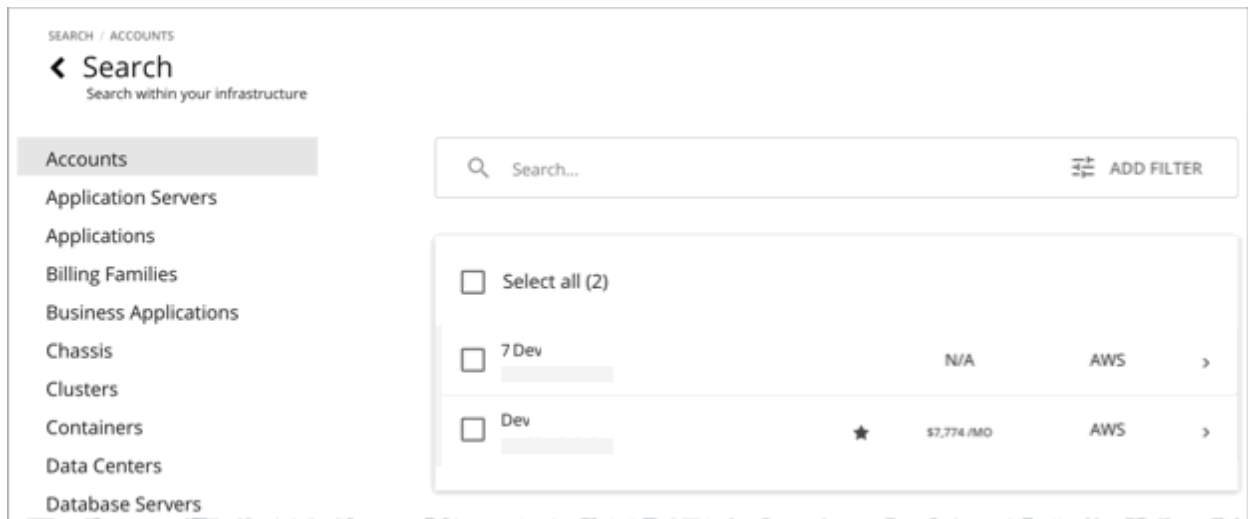
- From the Plan Page

Navigate to the Plan Page and click **NEW PLAN**. This plan has no scope. You will specify the scope after selecting the plan type.



- From the **Home Page**

To start a plan scenario from the **Home Page**, you must first go to the **Search** page to set the scope.



Set the scope to *a specific* Account, Billing Family, VM Group, or Region to start an Optimize Cloud plan.

- Cloud scope

If you set the scope to *a specific* Account, Billing Family, VM Group, or Region, you can start an Optimize Cloud or Buy VM Reservations plan.

- On-prem scope

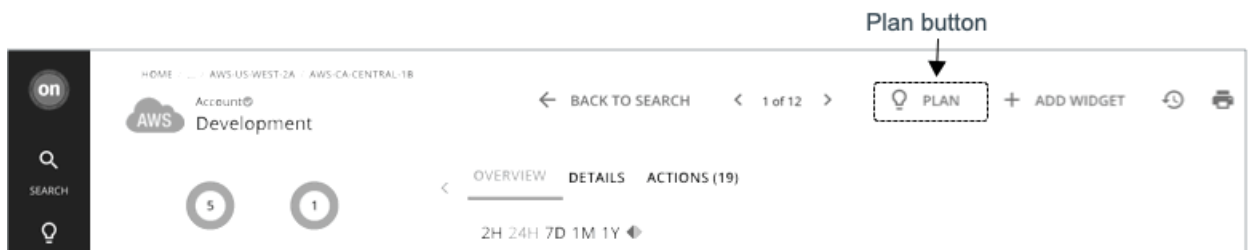
If you set the scope to *a specific* Cluster, Datacenter, Group, Storage Cluster, or Virtual Datacenter, you can start any plan. You may need to go through additional steps, depending on your chosen plan type. For example, if you scope to a cluster and choose the Add Virtual Machines plan type, the plan wizard prompts you to select the most suitable templates for the VMs you plan to add to the cluster.

- Container cluster scope

If you set the scope to *a specific* Container Platform Cluster, you can start an Optimize Container Cluster or Migrate Container Workloads plan.

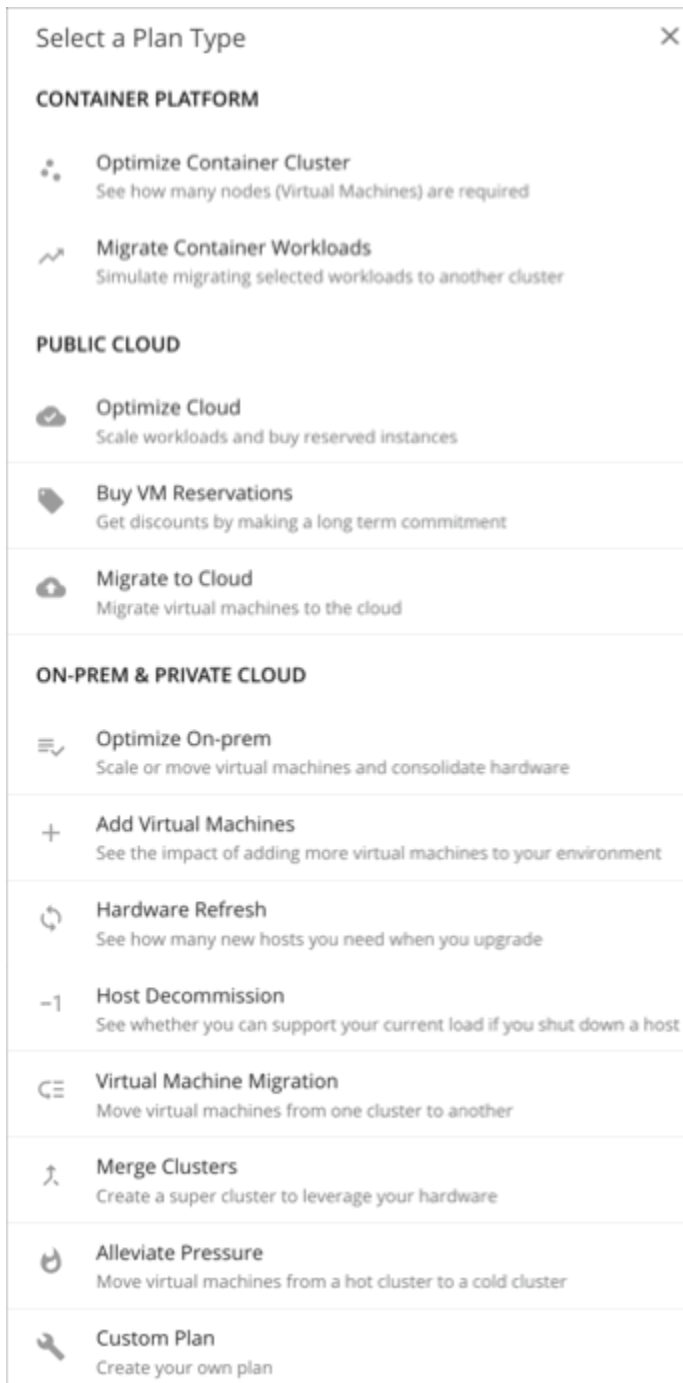
For details, see [Scoping the Turbonomic Session](#).

After setting the scope, the **Plan** button appears in the **Home Page**.



## 2. Plan Types

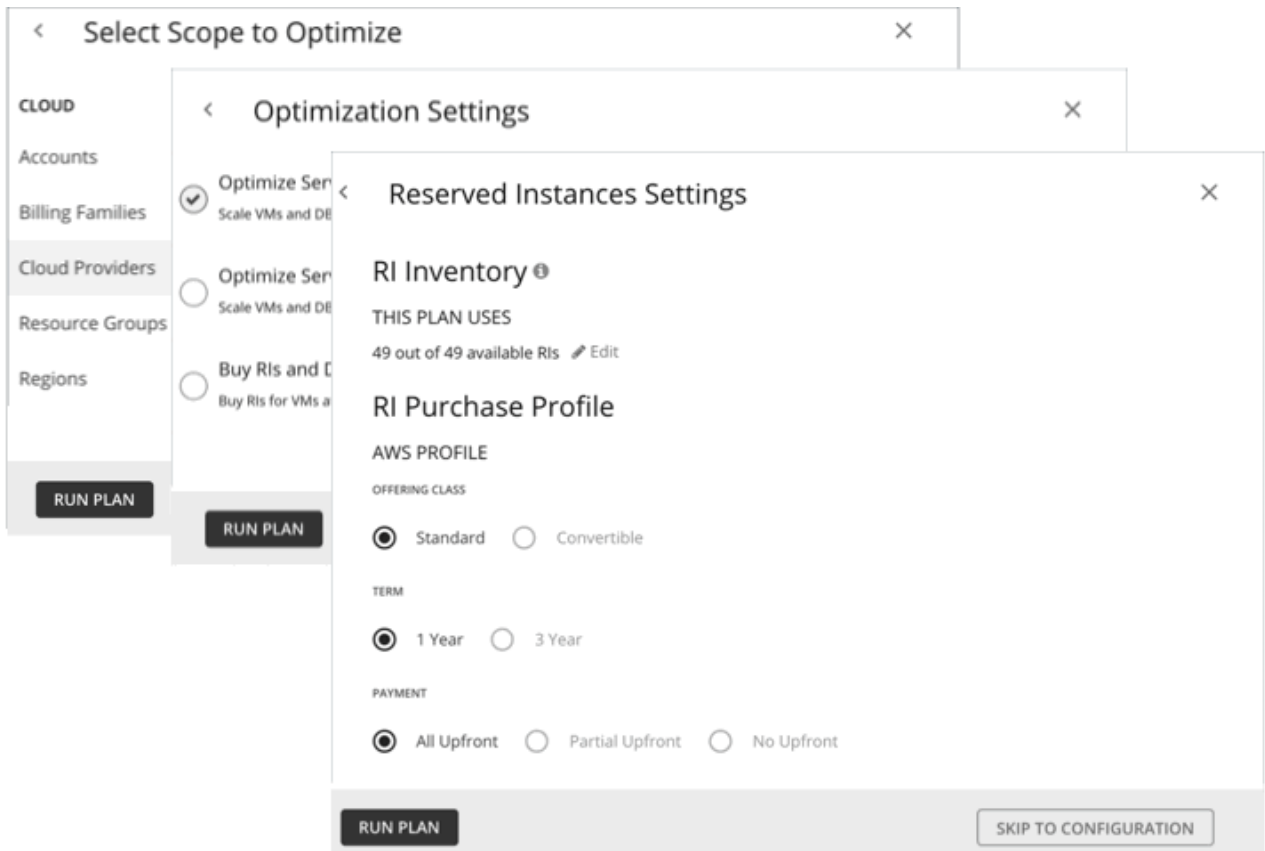
Select from the list of plan types. For more information, see [Plan Scenarios and Types](#).



Turbonomic opens the appropriate plan wizard.

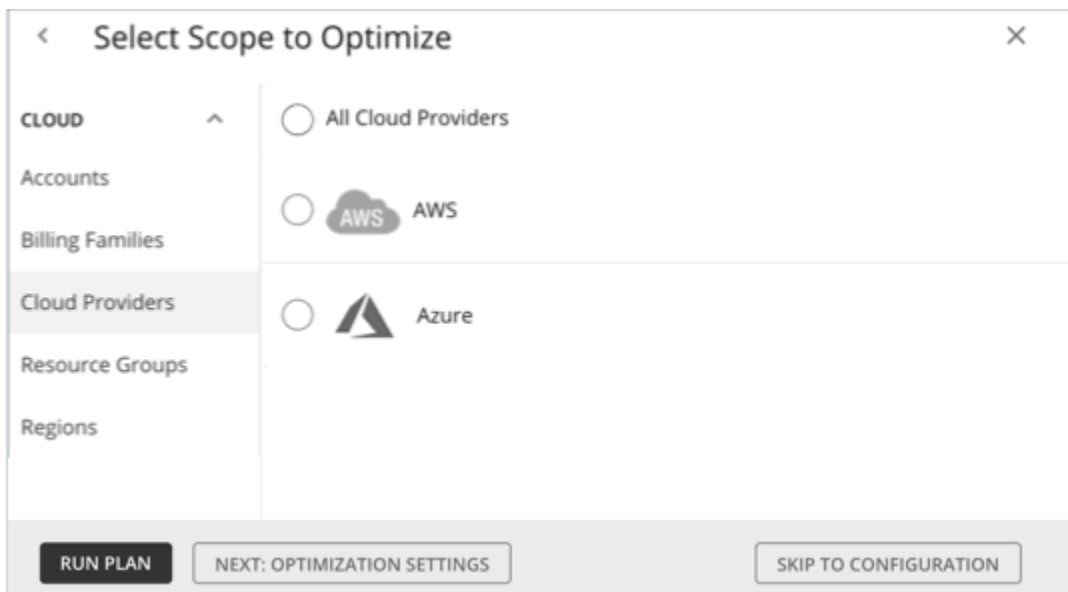
### 3. Plan Wizards

Each plan type includes a wizard to guide you through creating the scenario. The wizard leads you through the required configuration steps to create a plan that answers a specific question. After you make the required settings, you can skip ahead and run the plan, or continue through all the optional steps.



#### 4. Plan Scope

All plans require a scope. For example, to configure an Optimize Cloud plan, you set the scope to all or specific cloud providers or accounts.



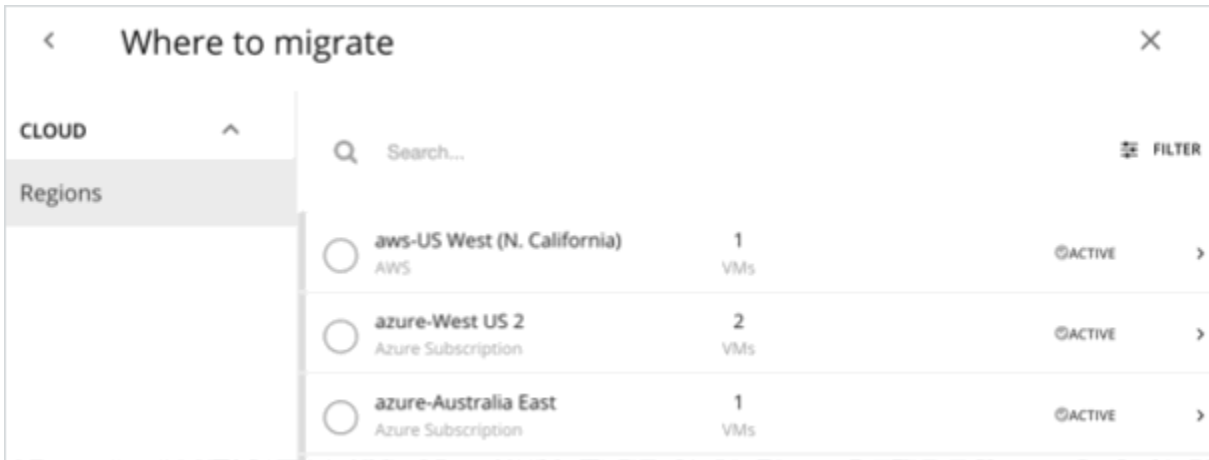
It usually helps to focus on a subset of your environment. For a very large environment, scoped plans run faster.

To narrow the scope, select a group from the list on the left side of the page. The page then refreshes to include only the entities belonging to that group.

Use **Search** or **Filter** to sort through a long list.

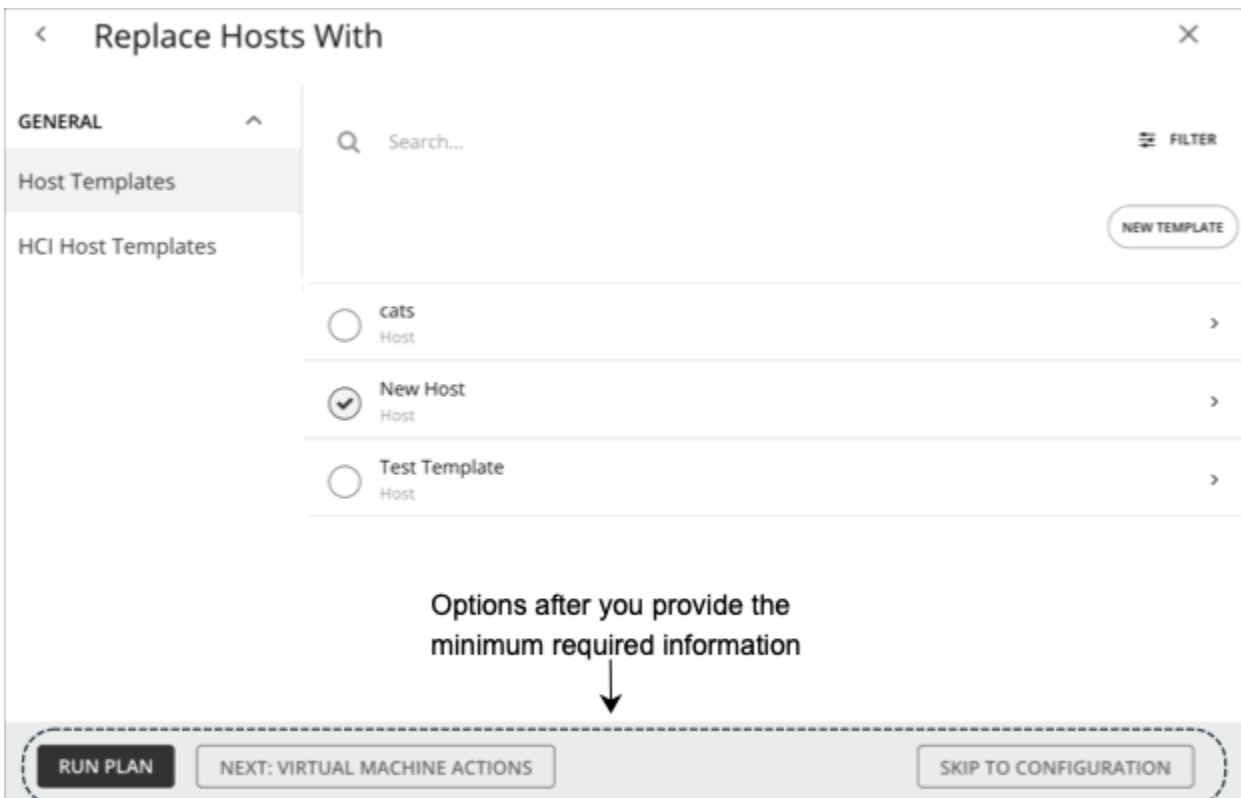
## 5. Additional Plan Information

The wizard prompts you for any additional information required to run the plan. For example, for a Hardware Refresh plan, you need to identify the hosts that will replace the scoped hosts. For a Migrate to Cloud plan, you need to identify the cloud service provider, region, or group you want the scoped workloads to migrate to.



## 6. Run the Plan

After you provide the minimum required information for running a plan, the wizard shows you the following options:



- **Run Plan:** Immediately run the plan.
- **Next: [Step]:** Continue with the rest of the wizard and then run the plan.
- **Skip to Configuration:** Skip the rest of the wizard and go to the Plan Page to:
  - Customize the plan settings.

- See a preview of the plan scenario.
- Run the plan.

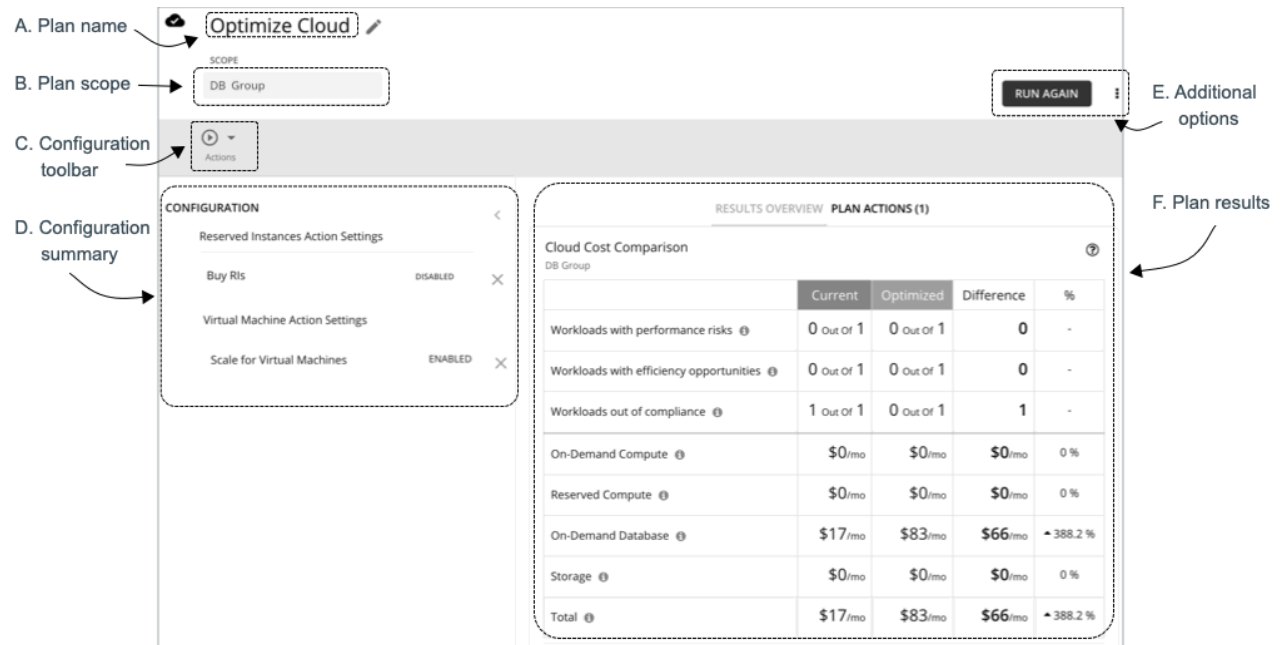
**Note:** For a custom plan, the only option available is **Configure Plan**. Click this button to open the Plan Page, configure the plan settings, and then run the plan.


## 7. The Plan Page


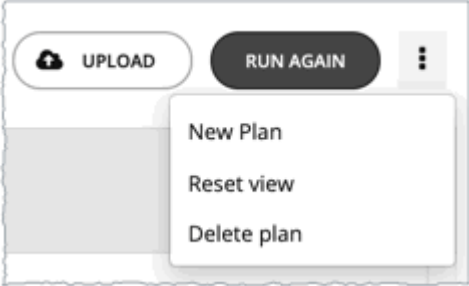
The Plan Page first displays if you skip the wizard or as soon as you run a plan.

For a plan with a large scope, it might take some time before you see the results. You can navigate away from the Plan Page and check the status in the Plan Management Page. You can also cancel a plan that is in progress.

The Plan Page shows the following sections:



| Plan Page Sections       | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| A. Plan name             | Turbonomic automatically generates a name when you create a new plan. Change the name to something that helps you recognize the purpose of this plan.                                                                                                                                                                                                                                                               |
| B. Plan scope            | Review the scope that you set in a previous step.<br><b>Note:</b><br>It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select <b>New Plan</b> . |
| C. Configuration toolbar | Configure additional settings for the plan. You can name the plan, change workload demand and the supply of resources, and specify other changes to the plan market. The toolbar items that display depend on the plan you are creating.                                                                                                                                                                            |

| Plan Page Sections       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| D. Configuration summary | <p>Review the plan's configuration settings. You can remove any setting by clicking the <b>x</b> mark on the right. Use the toolbar on top to change the settings. As you make changes to the plan scenario, those changes immediately appear in the Configuration summary.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| E. Additional options    | <p>See what else you can do with the plan.</p> <ul style="list-style-type: none"> <li>• <b>Upload:</b> (For Azure only) Upload the results of a Migrate to Cloud plan to the Azure Migrate portal. For details, see <a href="#">Uploading the Results to Azure</a>.</li> <li>• <b>Run / Run Again:</b> <ul style="list-style-type: none"> <li>– If a plan has not run, click <b>Run</b> and then check the plan results.</li> <li>– If the plan has run and you want to run it again with a different set of configuration settings, click <b>Run Again</b>. This runs the plan scenario against the market in its current state.</li> </ul> </li> <li>•  : Click to see more options.</li> </ul> <div data-bbox="894 926 1360 1209" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;">  </div> <ul style="list-style-type: none"> <li>– <b>New Plan:</b> Configure a new plan. You can choose this option if you want to change the scope of the current plan, which requires that you start over and configure a new plan.</li> <li>– <b>Reset view:</b> Restore charts to their default views. For example, if you changed the commodities displayed in the Optimized Improvements or Comparison charts, you can discard those changes by choosing this option.</li> <li>– <b>Delete plan:</b> Choose if you no longer need the plan.</li> </ul> |
| F. Plan results          | <p>Review the results in the charts provided.</p> <p>For a plan that has not run, you will see a <b>Scope Preview</b> chart and a one-time message instructing you to run the plan.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

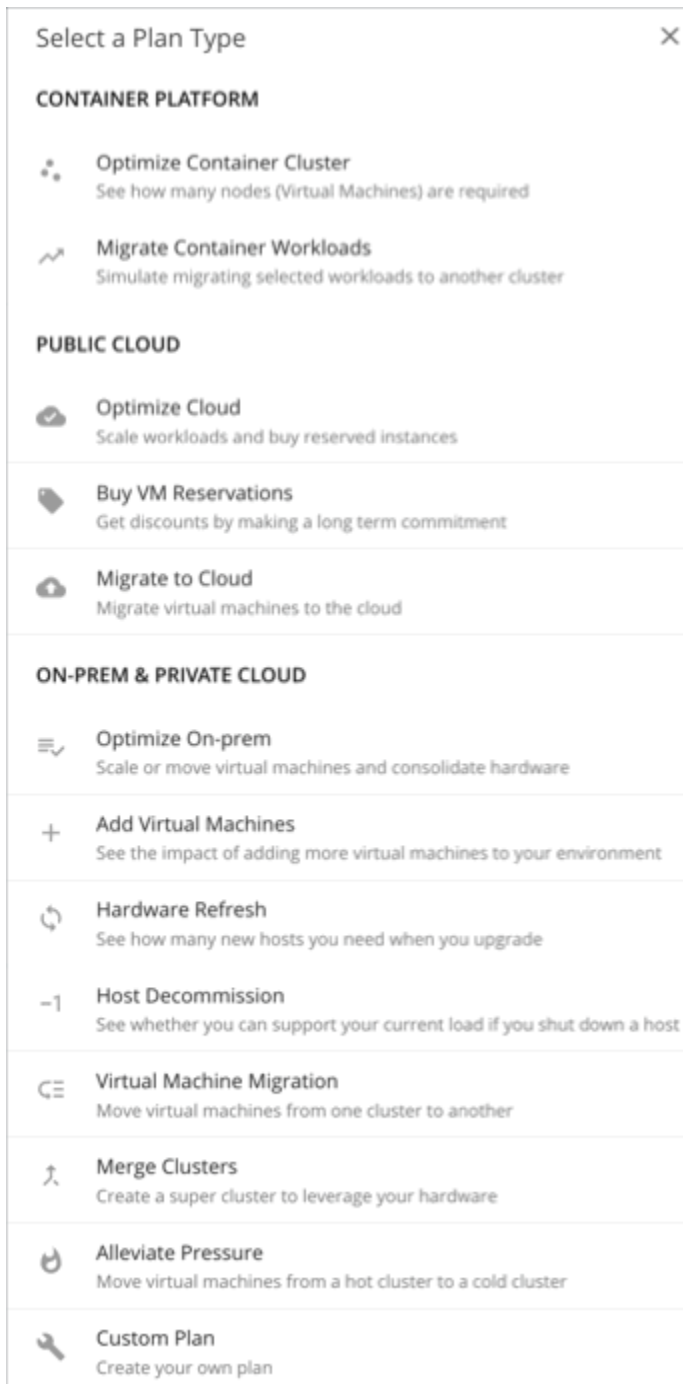
## 8. Plan Management

All the plans you have created display in the [Plan Management Page](#).



## Plan Scenarios and Types

To simulate different plan scenarios, Turbonomic provides the following general types of plans:



### Optimize Container Cluster

Run an Optimize Container Cluster plan to identify performance and efficiency opportunities for a single Kubernetes cluster. The results show the optimal number of nodes you need to assure performance for your existing workloads, and the impact of actions on the health of your container workloads and infrastructure.

## Migrate Container Workloads

Run a Migrate Container Workloads plan to simulate the migration of container workloads from one cluster to another. The plan compares results from a 'lift-and-shift only' scenario against a Turbonomic optimized plan. The results further highlight the actions you need to take to maintain and optimize workload performance in the new cluster.

## Optimize Cloud

For the scope of your public cloud environment that you want to examine, run a plan to see all the opportunities you have to reduce cost while assuring performance for your workloads. This includes suggestions to buy [discounts](#), comparisons of template and storage usage, and a comparison of current to optimized cost.

## Buy VM Reservations

Run the Buy VM Reservations plan to see the most cost-effective [discount](#) purchases that will continue to assure performance for your cloud VMs.

## Migrate to Cloud

A Migrate to Cloud plan simulates migration of on-prem VMs to the cloud, or migration of VMs from one cloud provider to another.

### Note:

For migrations within your on-prem environment, use the *Virtual Machine Migration* plan type.

## Optimize On-prem

See the effects of executing certain actions, such as scaling virtual machines, suspending hosts, or provisioning storage, to your on-prem environment.

## Add Virtual Machines

Adding virtual machines increases the demand that you place on your environment's infrastructure. You can set up a plan to add individual VMs or groups of VMs in your environment, or based on templates.

## Hardware Refresh

Choose hosts that you want to replace with different hardware. For example, assume you are planning to upgrade the hosts in a cluster. How many do you need to deploy, and still assure performance of your applications? Create templates to represent the upgraded hosts and let the plan figure out how many hosts you really need.

To increase the accuracy of the plan results, Turbonomic analysis considers a cluster's overall resource utilization over the last ten days. The platform identifies the day within those ten days when percentile utilization for the cluster reached 90%, and then uses each VM's actual utilization data *on that day* to perform its analysis.

### Note:

If you configure a Hardware Refresh plan to use a baseline snapshot, the plan will use that snapshot's data instead of the cluster's percentile data.

## Host Decommission

If your environment includes underutilized hardware, you can use a plan to see whether you can decommission hosts without affecting the workloads that depend on them.

## Virtual Machine Migration

Use this plan type to simulate workload migrations within your on-prem environment.

You can see whether you have enough resources to move your workload from its current provider group to another. For example, assume you want to decommission one datacenter and move all its workload to a different datacenter. Does the target datacenter have enough physical resources to support the workload move? Where should that workload be placed? How can you calculate the effect such a change would have on your overall infrastructure?

To calculate this information, create a plan that:

- Limits the plan scope to two datacenters (or clusters) – the one you will decommission, and the one that will take on the extra workload
- Removes all the hardware from the decommissioned datacenter
- Calculates workload placement across datacenter (or cluster) boundaries
- Does not provision new hardware to support the workload

## Merge Clusters

See the effects of merging two or more clusters. For example, you can see if merging the clusters would require provisioning additional storage to support current demand, or if ignoring cluster boundaries would improve performance and efficiency.

## Alleviate Pressure

Choose a cluster that shows bottlenecks or other risks to performance, and check to see the minimal changes you can make by migrating some workloads to another cluster. The cluster that is showing risks is a *hot* cluster, and the cluster you will migrate to is a *cold* cluster.

## Custom Plan

With a custom plan, you skip directly to the plan configuration after specifying the plan scope, and set up whatever type of scenario you want.

You would also choose **Custom Plan** if you need to run plans that include containers and container pods.

## Optimize Container Cluster Plan

The screenshot displays the 'Optimize Container Cluster 1' interface. The scope is set to 'Kubernetes-PT-AKS'. The configuration panel on the left shows three sections: 'Container Spec Action Settings' (Scale for ContainerSpec: ENABLED), 'Container Pod Action Settings' (Move for Container Pods: ENABLED), and 'Virtual Machines Action Settings' (Provision for Virtual Machines: ENABLED, Suspend for Virtual Machines: ENABLED, Scale for Virtual Machines: ENABLED). The main panel shows 'RESULTS OVERVIEW' and 'PLAN ACTIONS (258)'. The 'Optimize Container Cluster Summary' table is as follows:

|                               | Current    | After Plan | Difference | %        |
|-------------------------------|------------|------------|------------|----------|
| Container Pods                | 118        | 113        | 5          | ▼ 4.2 %  |
| Virtual Machines              | 8          | 7          | 1          | ▼ 12.5 % |
| Pod Density                   | 14.8 : 1   | 16.1 : 1   | 1.3 : 1    | ▲ 8.8 %  |
| Cluster CPU Capacity          | 20 Cores   | 22 Cores   | 2 Cores    | ▲ 10 %   |
| Cluster Memory Capacity       | 74 GB      | 82.9 GB    | 8.9 GB     | ▲ 12.1 % |
| Cluster Allocatable CPU       | 19.2 Cores | 21.1 Cores | 2 Cores    | ▲ 10.3 % |
| Cluster Allocatable Memory    | 53.8 GB    | 62.6 GB    | 8.8 GB     | ▲ 16.4 % |
| Cluster CPU Overcommitment    | 261 %      | 223.2 %    | 37.8 %     | ▼ 14.5 % |
| Cluster Memory Overcommitment | 83.7 %     | 45.4 %     | 38.3 %     | ▼ 45.8 % |

Run an Optimize Container Cluster plan to identify performance and efficiency opportunities for a single Kubernetes cluster. The results show the optimal number of nodes you need to assure performance for your existing workloads, and the impact of actions on the health of your container workloads and infrastructure. For example, you can see how container resize actions change the limits and requests allocated per namespace, or how node provision/suspend actions impact allocatable capacity for the cluster. For a cluster in the public cloud, the results also include the cost impact of node actions.

You can scope the plan to a:

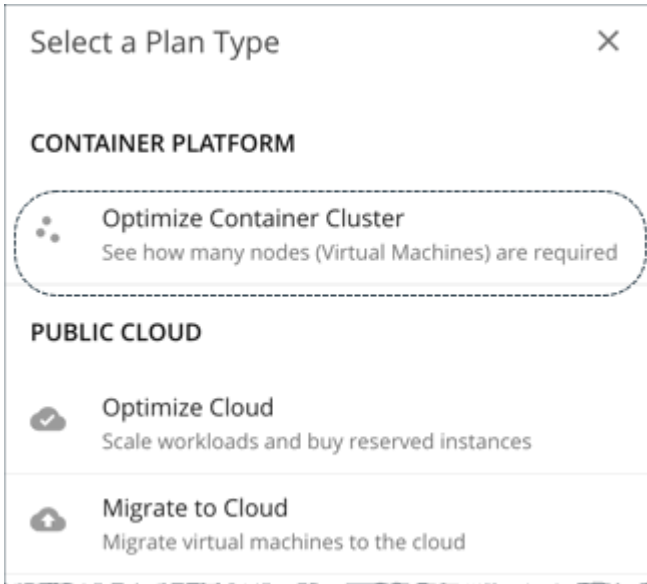
- Standalone container cluster
- Container cluster in an on-prem or public cloud environment
- Container cluster stitched to applications via Data Ingestion Framework (DIF)

Scoping to a group within a Kubernetes cluster (such as a group of nodes) is currently not supported.

### **Configuring an Optimize Container Cluster Plan**

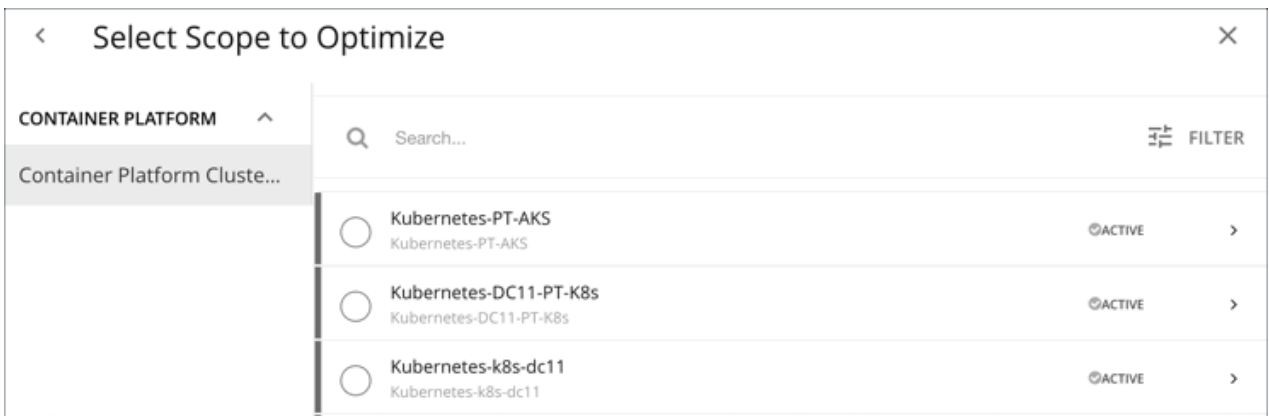
You can start an Optimize Container Cluster plan when you open the Plan page or set the scope to a Kubernetes cluster.

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).



## 1. Scope

Select a Kubernetes cluster to optimize.



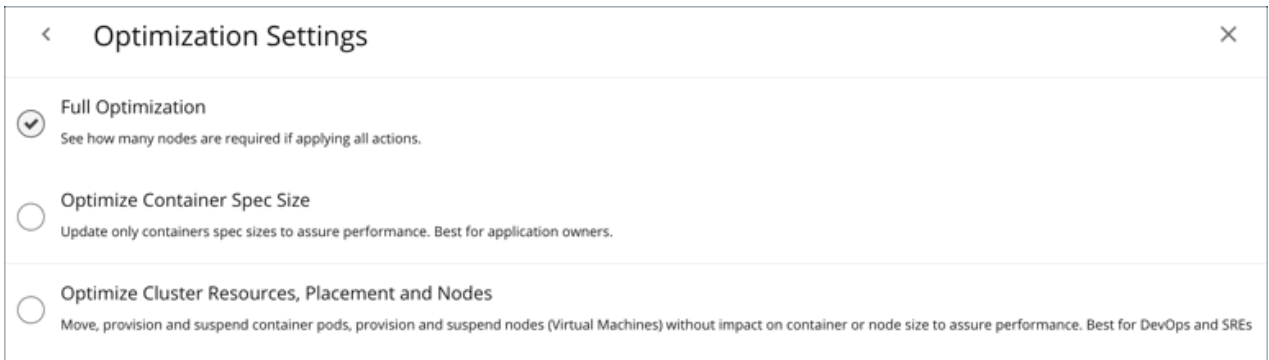
Scoping to a group within a Kubernetes cluster (such as a group of nodes) is currently not supported.

### Note:

After selecting a cluster, you can skip the next step (**Optimization Settings**) and run the plan. Turbonomic runs the **Full Optimization** scenario in this case.

## 2. Optimization Settings

Choose from the given optimization scenarios.



- **Full Optimization**

Turbonomic will recommend all relevant actions to optimize the cluster. For example, it can recommend provisioning nodes or resizing containers to meet application demand, or moving pods from one node to another to reduce congestion.

Turbonomic can recommend the following actions:

- Resize namespace compute resource quotas
- Resize container limits and requests
- Move pods
- Provision or suspend nodes
- Scale volumes

**Note:**

For a cluster in the public cloud, Turbonomic shows the cost impact of actions on nodes and volumes, to help you track your cloud spend. Turbonomic only reports the costs attached to these actions, and does *not* perform cost analysis on the cluster.

For a cluster in an on-prem environment, Turbonomic can also recommend the following actions:

- Move VMs
- Provision or suspend hosts
- Provision or suspend storage

- **Optimize Container Spec Size**

Turbonomic will only recommend resizing container limits and requests. This is ideal for application owners who manage the containers that their applications run on, but not the underlying container infrastructure.

- **Optimize Cluster Resources, Placement, and Nodes**

Turbonomic will recommend *all* relevant actions, *except* resizing container limits and requests. This is ideal for teams who oversee the health of your container infrastructure, and want to evaluate the impact of *not* rightsizing workloads.

After selecting an optimization scenario, you can:

- Run the plan.
- Or
- Choose **Skip to Configuration** to configure additional settings. See the next section for details.

## **(Optional) Additional Plan Settings**

You can fine tune your selected optimization scenario or include additional scenarios before you run the plan.

- Enable or disable actions

Fine tune your optimization scenario by enabling or disabling actions for containers, pods, or nodes. For example, you may have selected **Full Optimization**, but only for containers, nodes, and pods that are allowed to move. In this case, you would disable move actions for the pods that should never move.

For clusters in on-prem environments, you can also enable or disable actions for hosts and storage.

**Important:**

To avoid seeing inaccurate plan results, do *not* disable all actions.

- Add pods

See resource changes if you add more pods to the cluster. For example, you might need to provision nodes to accommodate the new pods.

Select an existing pod within or outside the selected Kubernetes cluster, and then specify how many copies to add. The plan simulates adding pods with the same resources as the selected pod.

- Remove pods or nodes

See the effect of removing pods or nodes from the cluster. For example, pod density could improve significantly if you remove pods that you no longer need, or certain pods might become unplaced if you remove nodes.

## Working with Optimize Container Cluster Plan Results

After the plan runs, you can view the results to see how the plan settings you configured affect your environment.

The screenshot displays the 'Optimize Container Cluster 1' interface. On the left, under 'CONFIGURATION', there are three sections: 'Container Spec Action Settings' (Scale for ContainerSpec: ENABLED), 'Container Pod Action Settings' (Move for Container Pods: ENABLED), and 'Virtual Machines Action Settings' (Provision for Virtual Machines: ENABLED, Suspend for Virtual Machines: ENABLED, Scale for Virtual Machines: ENABLED). On the right, the 'RESULTS OVERVIEW' tab is active, showing an 'Optimize Container Cluster Summary' table for 'Kubernetes-PT-AKS'.

|                               | Current    | After Plan | Difference | %        |
|-------------------------------|------------|------------|------------|----------|
| Container Pods                | 118        | 113        | 5          | ▼ 4.2 %  |
| Virtual Machines              | 8          | 7          | 1          | ▼ 12.5 % |
| Pod Density                   | 14.8 : 1   | 16.1 : 1   | 1.3 : 1    | ▲ 8.8 %  |
| Cluster CPU Capacity          | 20 Cores   | 22 Cores   | 2 Cores    | ▲ 10 %   |
| Cluster Memory Capacity       | 74 GB      | 82.9 GB    | 8.9 GB     | ▲ 12.1 % |
| Cluster Allocatable CPU       | 19.2 Cores | 21.1 Cores | 2 Cores    | ▲ 10.3 % |
| Cluster Allocatable Memory    | 53.8 GB    | 62.6 GB    | 8.8 GB     | ▲ 16.4 % |
| Cluster CPU Overcommitment    | 261 %      | 223.2 %    | 37.8 %     | ▼ 14.5 % |
| Cluster Memory Overcommitment | 83.7 %     | 45.4 %     | 38.3 %     | ▼ 45.8 % |

At the bottom right of the table area, there is a 'SHOW ALL >' link.

## General Guidelines

Familiarize yourself with these common terms that appear in the plan results:

- A container pod represents the compute demand from a running pod.
- A Kubernetes node (virtualized or bare metal) is represented as a VM.
- *Used* (or *Usage*) values represent actual resource consumption. For example, a node that consumes 100 MB of memory has a used value of 100 MB.
- *Utilization* values represent used/usage values against capacity. For example, a node that consumes 100 MB of memory against a total capacity of 500 MB has a utilization value of 20%.

## Optimize Container Cluster Summary

RESULTS OVERVIEW

PLAN ACTIONS (258)

|                               | Current    | After Plan | Difference | %        |
|-------------------------------|------------|------------|------------|----------|
| Container Pods                | 118        | 113        | 5          | ▼ 4.2 %  |
| Virtual Machines              | 8          | 7          | 1          | ▼ 12.5 % |
| Pod Density                   | 14.8 : 1   | 16.1 : 1   | 1.3 : 1    | ▲ 8.8 %  |
| Cluster CPU Capacity          | 20 Cores   | 22 Cores   | 2 Cores    | ▲ 10 %   |
| Cluster Memory Capacity       | 74 GB      | 82.9 GB    | 8.9 GB     | ▲ 12.1 % |
| Cluster Allocatable CPU       | 19.2 Cores | 21.1 Cores | 2 Cores    | ▲ 10.3 % |
| Cluster Allocatable Memory    | 53.8 GB    | 62.6 GB    | 8.8 GB     | ▲ 16.4 % |
| Cluster CPU Overcommitment    | 261 %      | 223.2 %    | 37.8 %     | ▼ 14.5 % |
| Cluster Memory Overcommitment | 83.7 %     | 45.4 %     | 38.3 %     | ▼ 45.8 % |

[SHOW ALL >](#)

This chart shows how your container environment and the underlying resources will change after you execute the actions that the plan recommends. The chart shows the following information:

- **Container Pods**

Count of active container pods in the plan.

- **Virtual Machines**

Count of active nodes in the plan. This chart does not count "non-participating" entities in the real-time market, such as suspended nodes.

- **Pod Density**

Average number of pods per node.

For the total number of pods against the node capacity (maximum pods per node), see the **Number of Consumers** data in the following charts:

- Nodes (VMs) Optimized Improvements
- Nodes (VMs) Comparison
- Container Cluster Optimized Improvements
- Container Cluster Comparison

- **Cluster CPU Capacity**

Total CPU capacity for the cluster. The 'After Plan' result indicates how much CPU capacity will result in the optimal number of nodes required to run workloads.

- **Cluster Memory Capacity**



Total memory capacity for the cluster. The 'After Plan' result indicates how much memory capacity will result in the optimal number of nodes required to run workloads.

- **Cluster Allocatable CPU**

Total amount of cluster CPU available for pod requests. The 'After Plan' result indicates how much of the allocatable CPU capacity will change if you provision or suspend nodes.

- **Cluster Allocatable Memory**

Total amount of cluster memory available for pod requests. The 'After Plan' result indicates how much of the allocatable memory capacity will change if you provision or suspend nodes.

- **Cluster CPU Overcommitment**

(Only for containers with CPU limits) This indicates whether the CPU limits exceed the capacity of the underlying nodes. A value greater than 100% indicates overcommitment. Turbonomic manages cluster resources by actual utilization and limit rightsizing so that you can run more workloads with less risk.

Turbonomic only calculates overcommitment in plans. The calculation can be expressed as:

$$\text{Overcommitment} = \text{Sum of CPU limits for all containers} / \text{Sum of CPU capacity for all nodes}$$

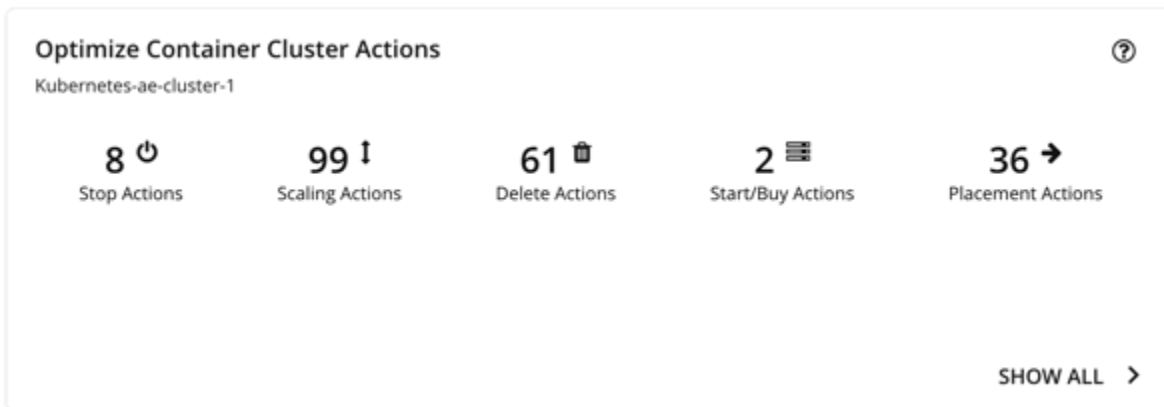
- **Cluster Memory Overcommitment**

(Only for containers with memory limits) This indicates whether the memory limits exceed the capacity of the underlying nodes. A value greater than 100% indicates overcommitment. Turbonomic manages cluster resources by actual utilization and limit rightsizing so that you can run more workloads with less risk.

Turbonomic only calculates overcommitment in plans. The calculation can be expressed as:

$$\text{Overcommitment} = \text{Sum of memory limits for all containers} / \text{Sum of memory capacity for all nodes}$$

## Optimize Container Cluster Actions



This chart summarizes the actions that you need to execute to achieve the plan results. For example, you might need to resize limits and requests for containers (via the associated Workload Controllers) to address performance issues. Or, you might need to move pods from one node to another to reduce congestion.

Smarter redistribution and workload rightsizing also drive cluster optimization, resulting in the need to provision node(s) based on application demand, or to defragment node resources to enable node suspension.

Turbonomic can recommend the following actions:

- Resize namespace compute resource quotas
- Resize container limits and requests

**Note:**

Executing several container resize actions can be very disruptive since pods need to restart with each resize. For replicas of the container scale group(s) related to a single Workload Controller, Turbonomic consolidates resize actions into one *merged action* to minimize disruptions. When a merged action has been executed (via the associated Workload Controller), all resizes for all related container specifications will be changed at the same time, and pods will restart once.

- Move pods
- Provision or suspend nodes
- Scale volumes

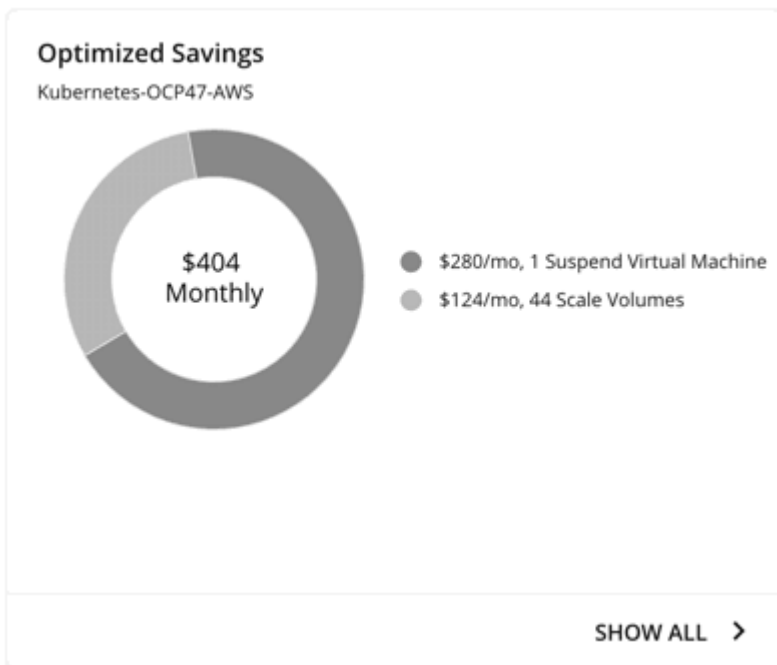
**Note:**

For a cluster in the public cloud, Turbonomic shows the cost impact of actions on nodes and volumes, to help you track your cloud spend. Turbonomic only reports the costs attached to these actions, and does *not* perform cost analysis on the cluster. See the Optimized Savings and Optimized Investments charts for more information.

For an on-prem cluster, Turbonomic can also recommend the following actions:

- Move VMs
- Provision or suspend hosts
- Provision or suspend storage

### Optimized Savings



For a cluster in the public cloud, Turbonomic shows the savings you would realize if you execute the actions (such as node suspension) that the plan recommends to increase infrastructure efficiency. Note that efficiency is the driver of this action, *not* cost. Cost information is included to help you track your cloud spend.

The chart shows total monthly savings. Click **Show All** to view the actions with cost savings.

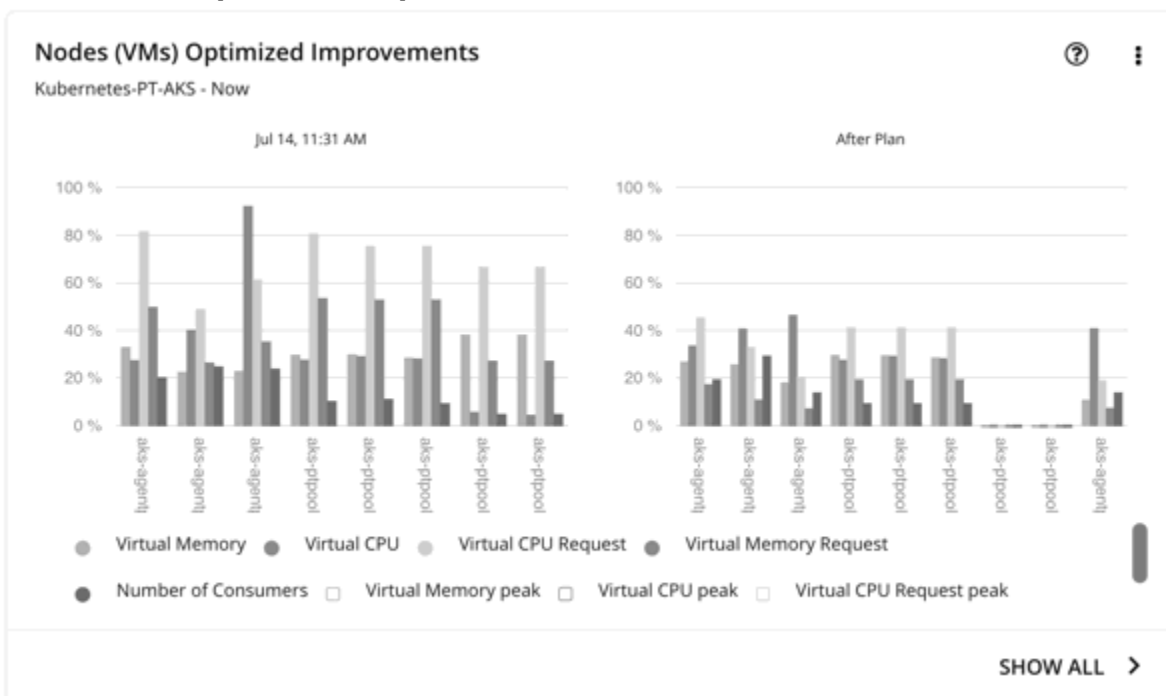
## Optimized Investments



For a cluster in the public cloud, Turbonomic shows the costs you would incur if you execute the node and volume scaling actions that the plan recommends to address performance issues. For example, if some applications risk losing performance, Turbonomic can recommend provisioning nodes to increase capacity. This chart shows how these actions translate to an increase in expenditure. Note that performance and efficiency are the drivers of these actions, *not* cost. Cost information is included to help you plan for the increase in capacity.

The chart shows total monthly investments. Click **Show All** to view the actions that require investments.

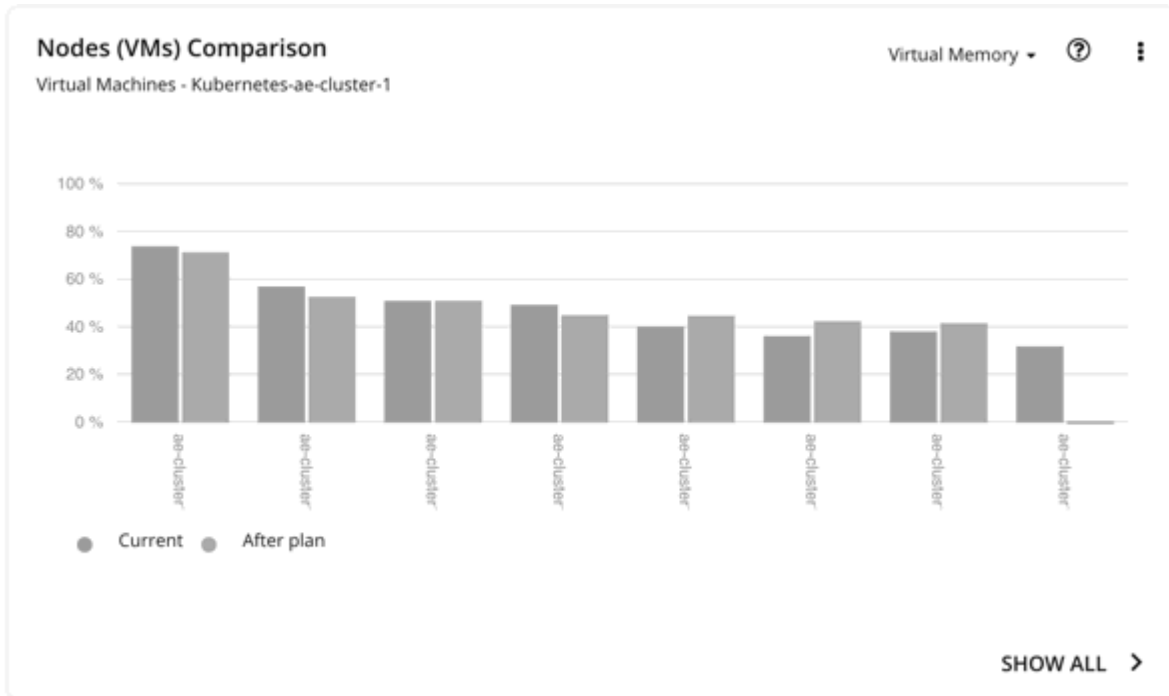
## Nodes (VMs) Optimized Improvements



This chart compares the following before and after the plan:

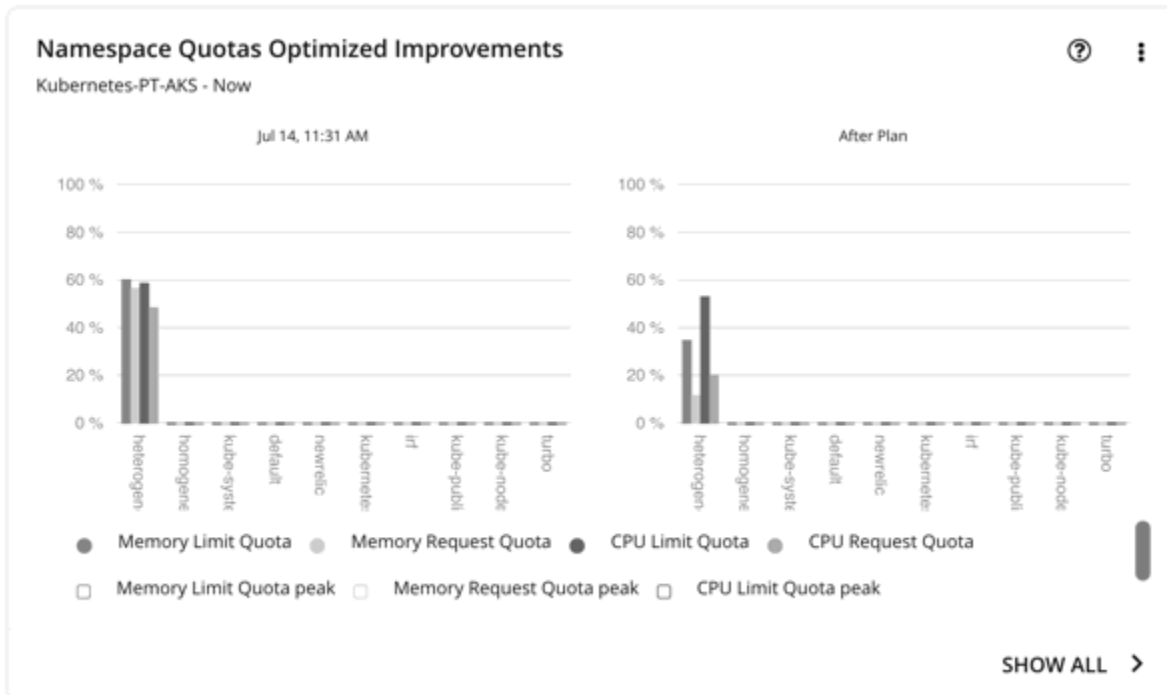
- Utilization of the following for all nodes:
  - vMem
  - vCPU
  - vMem Request
  - vCPU Request
- Number of pods consuming resources against the maximum pod capacity for all the nodes

## Nodes (VMs) Comparison



This chart compares node resource utilization (one metric at a time) before and after the plan.

## Namespace Quotas Optimized Improvements



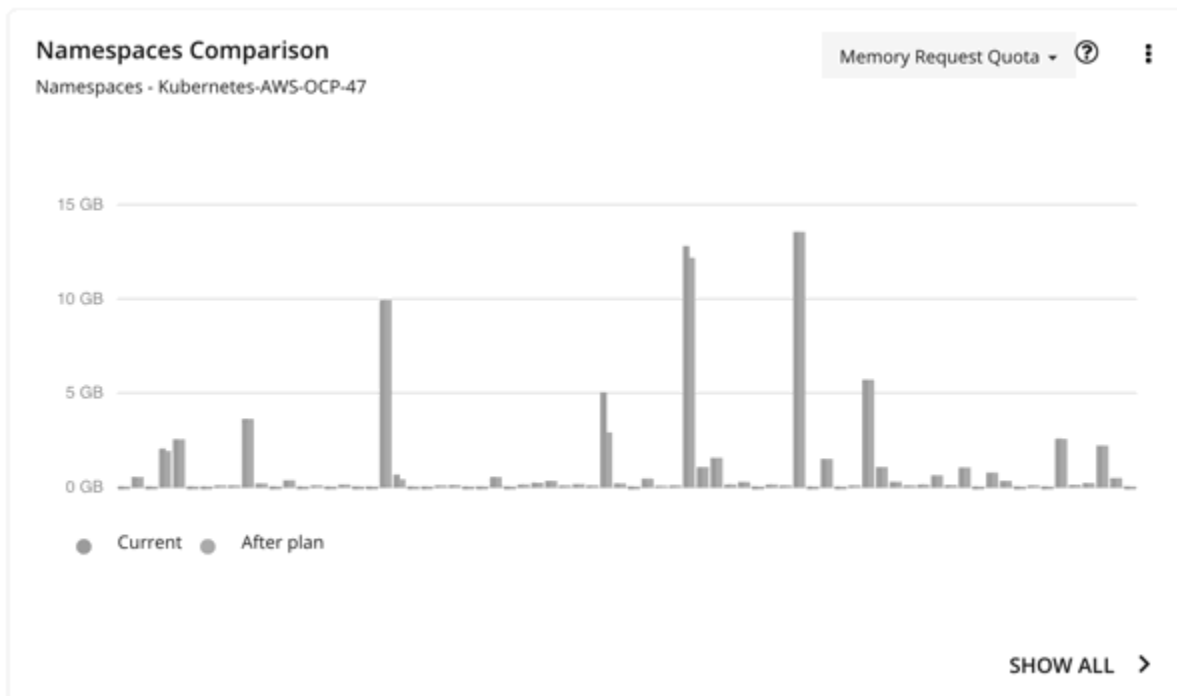
This chart shows pod utilization of resource quotas defined in namespaces. Resource quotas include:

- CPU Limit Quota
- Memory Limit Quota
- CPU Request Quota
- Memory Request Quota

For namespaces without defined quotas, utilization is 0 (zero).

With or without quotas, you can see the sum of pod limits and requests per namespace. Go to the top-right section of the Plan Results page, click the download button, and select **Namespace**. Utilization data in the downloaded file shows these limits and requests. You can also compare usage values in the Namespaces Comparison chart.

## Namespaces Comparison



This chart compares namespace quota usage (one metric at a time) before and after the plan.

Use this chart to see how container resizing changes the limits and requests allocated per namespace, whether you leverage quotas or not.

To achieve the 'After Plan' results, click **Show All**. In the Details page that opens, go to the Name column and then click the namespace link. This opens another page with a list of pending actions for the namespace.

### Namespaces Comparison

| Name            | Current            |                         |                    | After Plan         |                         |                    |
|-----------------|--------------------|-------------------------|--------------------|--------------------|-------------------------|--------------------|
|                 | Memory Limit Quota | Memory Limit Quota Used | Memory Limit Quota | Memory Limit Quota | Memory Limit Quota Used | Memory Limit Quota |
| turbonomic      | ∞                  | 570.7 GB                | 0 %                | ∞                  | 70.3 GB                 | 0 %                |
| kube-node-lease | ∞                  | 0 KB                    | 0 %                | ∞                  | 0 KB                    | 0 %                |

Namespace **turbonomic**

OVERVIEW | DETAILS | POLICIES | **ACTIONS (47)**

**RESIZE** ^

Workloa... (46)

**PROVISI...** ^

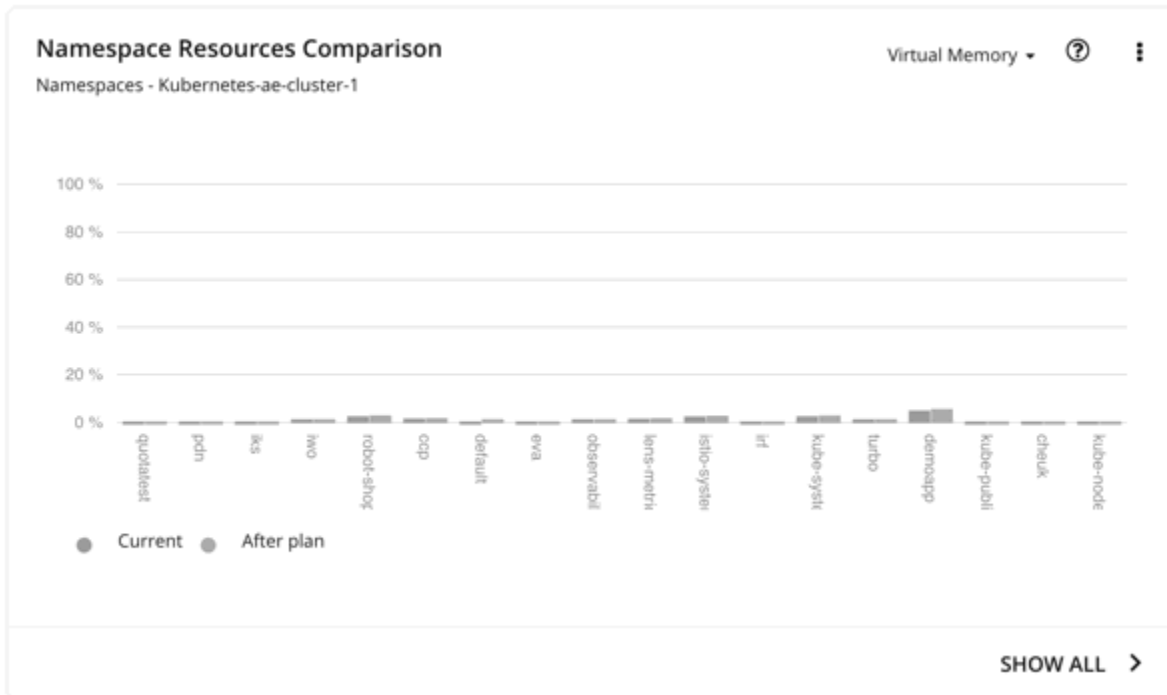
Virtual Ma... (1)

**Resize Actions (46)**

Type to search

- Workload Controller Name: Container Cluster
- mediation-awscost: Kubernetes-Turbonomic
- mediation-aws: Kubernetes-Turbonomic

## Namespace Resources Comparison



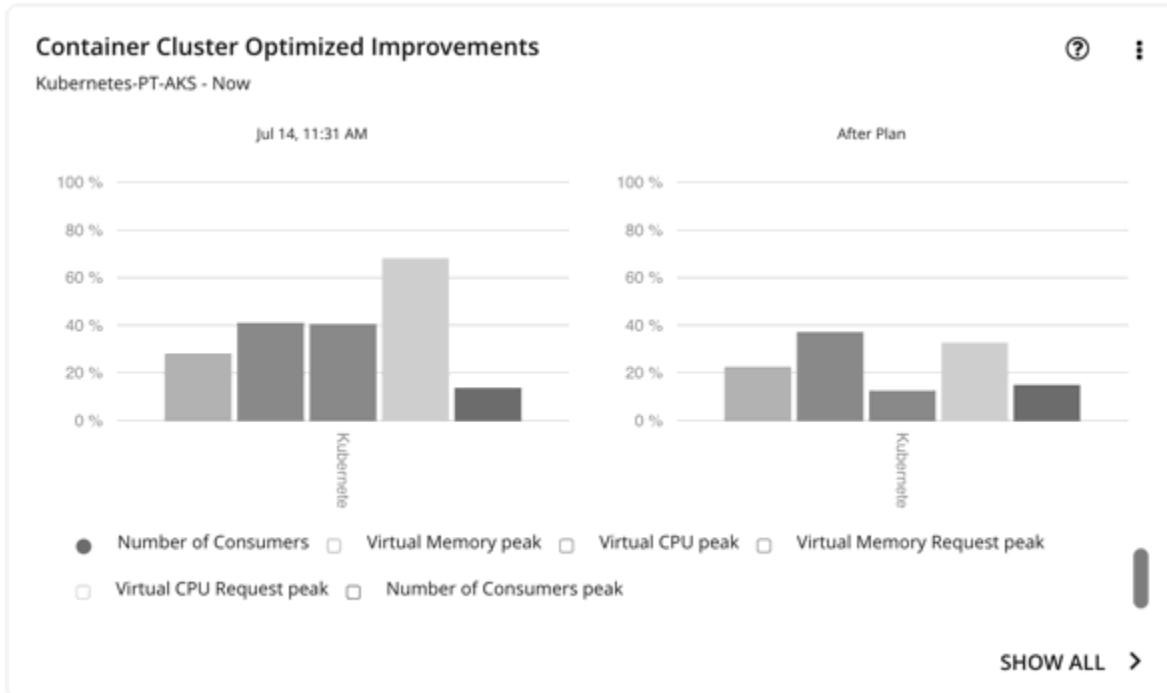
This chart shows how much cluster resources per namespace are utilized by pods. Utilization can be expressed as follows:

$$\text{Utilization} = \frac{\text{Sum of actual vMem/vCPU used by pods}}{\text{vMem/vCPU capacity for the cluster}}$$

This information helps you understand which namespaces use the most cluster resources. You can also use it for showback analysis. vMem and vCPU utilized by pods in the namespaces would change when the number of nodes changes as a result of executing the plan actions.

This chart is especially useful if you do not have resource quotas defined in your namespaces.

## Container Cluster Optimized Improvements



This chart shows the following, assuming you execute all actions in the plan:

- Changes to the utilization of cluster resources
- Overcommitment values



## Container Cluster Comparison



This chart compares the following before and after the plan:

- Utilization of cluster resources (one metric at a time)
- Overcommitment values

### Optimized Improvements for Hosts, Storage, and Virtual Machines

Use these charts if you ran the plan on an on-prem Kubernetes cluster. These charts show how the utilization of resources would change assuming you accept all of the actions listed in the Plan Actions chart

### Hosts, Storage Devices, and Virtual Machines Comparison

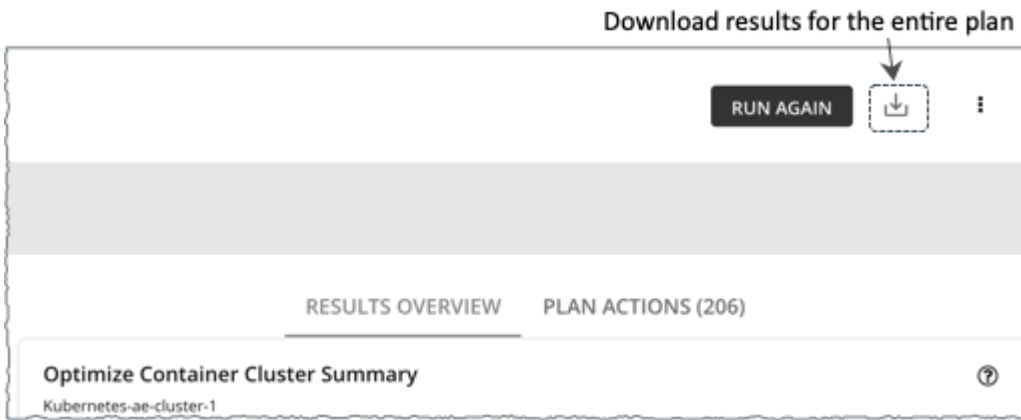
Use these charts if you ran the plan on an on-prem Kubernetes cluster. These charts show how the utilization of a particular commodity (such as memory or CPU) for each entity in the plan would change if you execute the recommended actions.

#### Note:

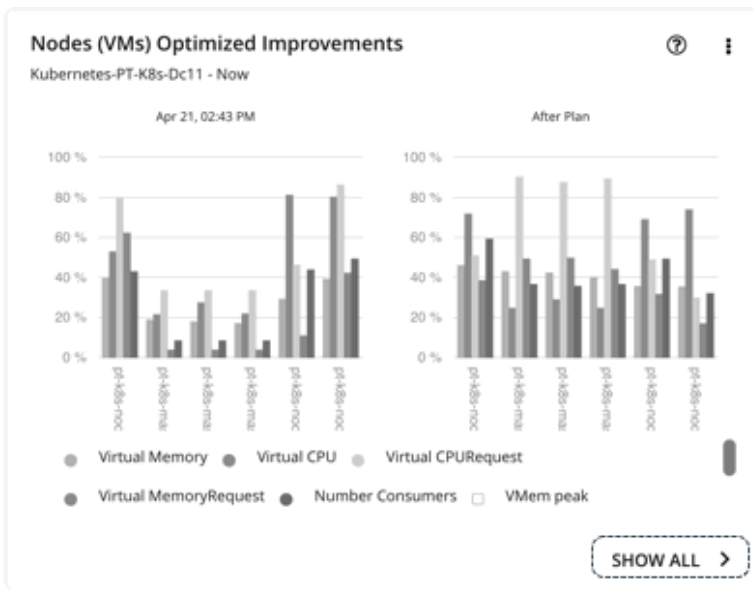
For the Storage Devices Comparison chart, if you set the view to **VM Per Storage** and click **Show all**, the total number of VMs sometimes does not match the number in the Summary chart. This happens if there are VMs in the plan that use multiple storage devices. The Storage Devices Comparison chart counts those VMs multiple times, depending on the number of storage devices they use, while the Plan Summary chart shows the actual number of VMs.

### Downloading Plan Results

To download results for nodes, namespaces, or the container cluster, click the download button at the top-right section of the Plan Results page.



You can also download the plan results shown in individual charts. Click the **Show All** button for a chart, and then the download button at the top-right section of the Details page.



| Virtual Machine Name | Container Cluster    | Number Consumers | Number Consumers | Virtual CPU | Virtual CPU | Virtual CPURequest | Virtual CPURequest |
|----------------------|----------------------|------------------|------------------|-------------|-------------|--------------------|--------------------|
| pt-k8s-node-2        | Kubernetes-PT-K8s... | 49.1 %           | 49.1 %           | 80 %        | 82.1 %      | 86.1 %             | 86.1 %             |
| pt-k8s-node-1        | Kubernetes-PT-K8s... | 42.7 %           | 42.7 %           | 52.7 %      | 59.5 %      | 79.5 %             | 79.5 %             |


For charts that display infinite capacities (for example, the Namespaces Comparison chart), the downloaded file shows an unusually high value, such as 1,000,000,000 cores, instead of the  $\infty$  symbol.

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.

**Note:**

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Migrate Container Workloads Plan

Run a Migrate Container Workloads plan to simulate the migration of container workloads from one cluster to another. For example, when you are ready to deploy workloads in your test cluster to production, you can run this plan to evaluate resource requirements in the production cluster. You might also have a need to decommission an existing cluster or consolidate workloads in different clusters into a single cluster, both of which require moving existing workloads to a new cluster. Run this plan to see if all workloads can be placed in the new cluster.

The destination cluster (i.e., the cluster that you choose for migration) can be a:

- Standalone container cluster
- Container cluster in an on-prem or public cloud environment. For cloud environments, cost information is included in the plan results.
- Container cluster stitched to applications via Data Ingestion Framework (DIF)

The plan results show the impact of workload migration on the destination cluster, based on two migration scenarios:

|                               | BEFORE MIGRATION | LIFT & SHIFT | OPTIMIZED    |
|-------------------------------|------------------|--------------|--------------|
| Workload Controllers          | 69               | 71           | 71           |
| Container Pods                | 141              | 145          | 145          |
| Virtual Machines              | 7                | 7            | 5            |
| Pod Density                   | 20.1 : 1         | 20.7 : 1     | 29 : 1       |
| Cluster CPU Capacity          | 17 Cores         | 27 Cores     | 17 Cores     |
| Cluster CPU Allocatable       | 17 Cores         | 17 Cores     | 17 Cores     |
| Cluster CPU Request           | 4.6 Cores        | 4.6 Cores    | 3.85 Cores   |
| Cluster CPU Limit             | 773.81 Cores     | 269.81 Cores | 270.22 Cores |
| Cluster CPU Overcommitment    | 4551.8 %         | 999.3 %      | 1589.5 %     |
| Cluster Memory Capacity       | 39.94 GB         | 61.94 GB     | 39.94 GB     |
| Cluster Memory Allocatable    | 39.45 GB         | 39.45 GB     | 39.45 GB     |
| Cluster Memory Request        | 5.82 GB          | 5.83 GB      | 5.83 GB      |
| Cluster Memory Limit          | 27.72 GB         | 22.05 GB     | 15.81 GB     |
| Cluster Memory Overcommitment | 69.4 %           | 35.6 %       | 39.6 %       |

[SHOW ALL >](#)

- Lift & Shift

The Lift & Shift scenario migrates your container workloads based on the resources currently available in the destination cluster.

- Optimized

The Optimized scenario identifies opportunities to optimize performance in the destination cluster. For example, after analyzing historical resource utilization, Turbonomic might recommend resizing limits and requests for containers (via the associated Workload Controllers) to maintain performance. If you were to migrate workloads based on the resources currently available in the destination cluster, then your applications could risk losing performance.

**Note:**

The plan results do not show data for the source cluster.

The results further highlight the actions you need to take to maintain and optimize workload performance in the destination cluster.

### **Configuring a Migrate Container Workloads Plan**

You can begin creating a plan scenario from two places in the user interface:

- From the Plan Page (Recommended)

On the left menu, click **Plan** to open the Plan Page. Click **New Plan**, and then select **Migrate Container Workloads**.

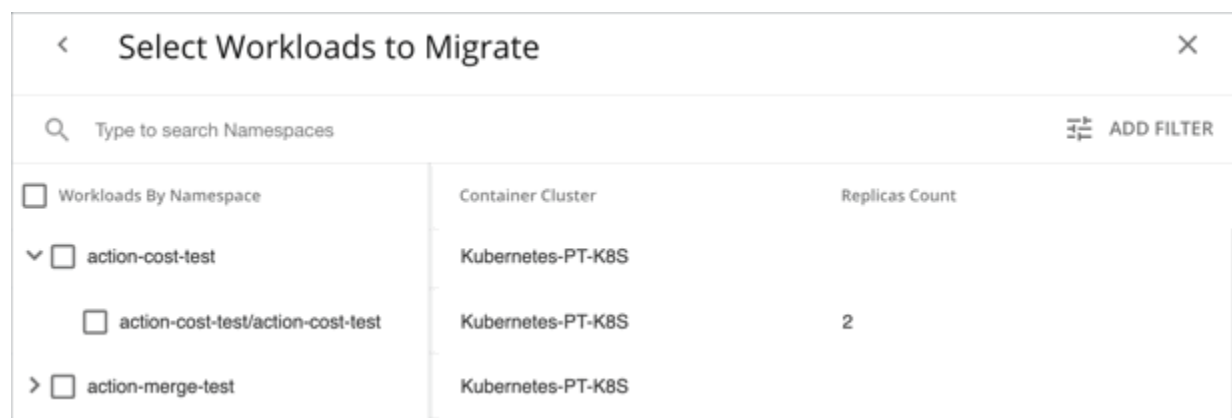
- From the supply chain

To start a plan scenario from the supply chain, set the scope to a container cluster and then click **Plan** at the top-right corner of the page.

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

## **1. Scope**

Select the container workloads that you want to migrate.



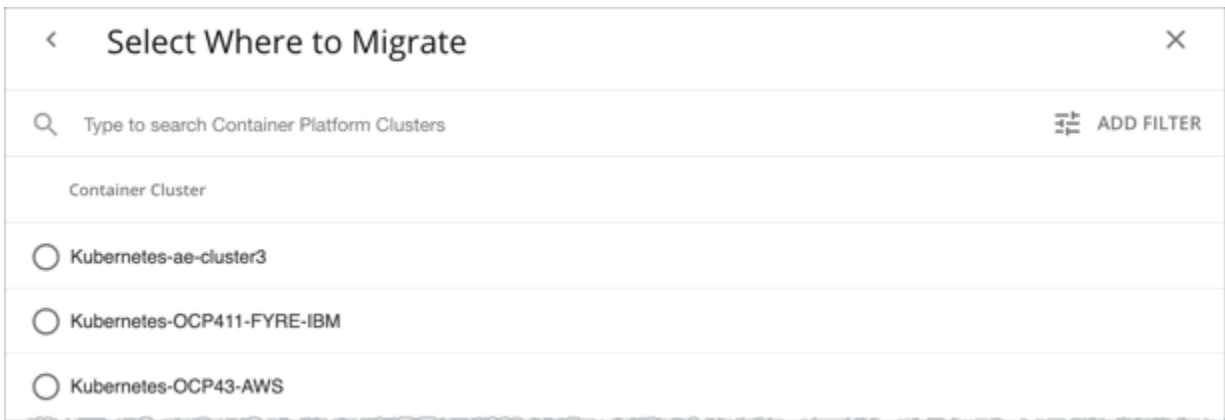
Container workloads are grouped by namespace. Expand a namespace to see individual Workload Controllers that manage workloads, and the number of container replicas for each Workload Controller. You can adjust the number of replicas in a later step. For example, you can add replicas to meet workload demand.

To define your scope, select individual namespaces and/or Workload Controllers, from one or several clusters. For example, if you are migrating workloads in two clusters, select the namespaces in those clusters.

To find specific workloads, type a keyword in the Search bar or click **Add Filter**. You can filter workloads by container cluster, namespace, tag, or Workload Controller.

## **2. Where to Migrate**

Choose the destination cluster for the container workloads you selected.

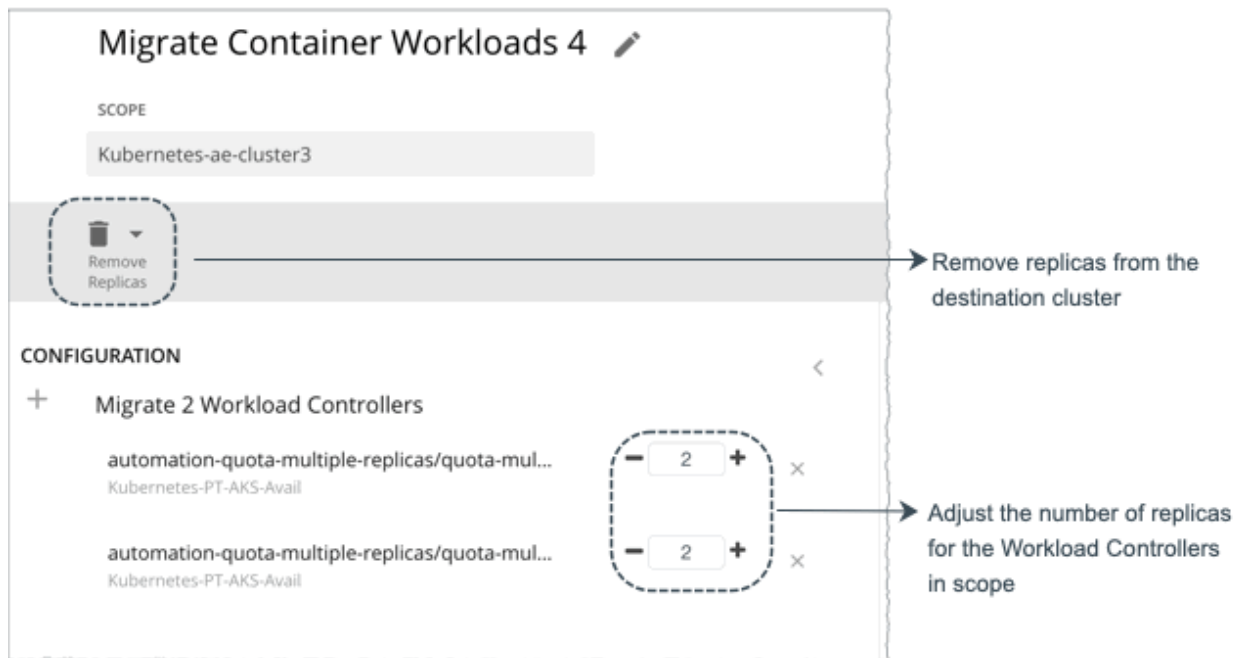


You can migrate to a:

- Standalone container cluster
- Container cluster in an on-prem or public cloud environment
- Container cluster stitched to applications via Data Ingestion Framework (DIF)

After selecting a cluster, you can:

- Run the plan.
- Or
- Choose **Skip to Configuration** to configure additional settings, including:



- Removing existing replicas from the destination cluster. You may need to do this to make room for the workloads you are migrating.
- Adjusting the number of replicas for the Workload Controllers in scope. For example, you may need to add replicas to meet workload demand.

### **Working with Migrate Container Workloads Plan Results**

After the plan runs, you can view the results to see the impact of workload migration on the *destination* cluster. The results do *not* show data for the *source* cluster (i.e., the cluster that currently hosts the workloads).

## General Guidelines

Familiarize yourself with these common terms that appear in the plan results:

- A container pod represents the compute demand from a running pod.
- A Kubernetes node (virtualized or bare metal) is represented as a VM.
- *Used* (or *Usage*) values represent actual resource consumption. For example, a node that consumes 100 MB of memory has a used value of 100 MB.
- *Utilization* values represent used/usage values against capacity. For example, a node that consumes 100 MB of memory against a total capacity of 500 MB has a utilization value of 20%.

## Migrate Container Workloads Summary

|                               | BEFORE MIGRATION | LIFT & SHIFT | OPTIMIZED    |
|-------------------------------|------------------|--------------|--------------|
| Workload Controllers          | 69               | 71           | 71           |
| Container Pods                | 141              | 145          | 145          |
| Virtual Machines              | 7                | 7            | 5            |
| Pod Density                   | 20.1 : 1         | 20.7 : 1     | 29 : 1       |
| Cluster CPU Capacity          | 17 Cores         | 27 Cores     | 17 Cores     |
| Cluster CPU Allocatable       | 17 Cores         | 17 Cores     | 17 Cores     |
| Cluster CPU Request           | 4.6 Cores        | 4.6 Cores    | 3.85 Cores   |
| Cluster CPU Limit             | 773.81 Cores     | 269.81 Cores | 270.22 Cores |
| Cluster CPU Overcommitment    | 4551.8 %         | 999.3 %      | 1589.5 %     |
| Cluster Memory Capacity       | 39.94 GB         | 61.94 GB     | 39.94 GB     |
| Cluster Memory Allocatable    | 39.45 GB         | 39.45 GB     | 39.45 GB     |
| Cluster Memory Request        | 5.82 GB          | 5.83 GB      | 5.83 GB      |
| Cluster Memory Limit          | 27.72 GB         | 22.05 GB     | 15.81 GB     |
| Cluster Memory Overcommitment | 69.4 %           | 35.6 %       | 39.6 %       |

[SHOW ALL >](#)

### Table Columns

Table columns shows the following information for the destination cluster:

- Before Migration

This column shows the state of the destination cluster before workload migration.

- Lift & Shift

The Lift & Shift scenario migrates your container workloads based on the resources currently available in the destination cluster.

- Optimized

The Optimized scenario identifies opportunities to optimize performance in the destination cluster. For example, after analyzing historical resource utilization, Turbonomic might recommend resizing limits and requests for containers (via the associated Workload Controllers) to maintain performance. If you were to migrate workloads based on the resources currently available in the destination cluster, then your applications could risk losing performance.

## Table Rows

Table rows show the following information for the destination cluster:

| Row                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Workload Controllers    | Count of active Workload Controllers. This chart does not count "non-participating" entities in the real-time market, such as inactive Workload Controllers.                                                                                                                                                                                                                                                                                        |
| Container Pods          | Count of active (running) container pods.                                                                                                                                                                                                                                                                                                                                                                                                           |
| Virtual Machines        | Count of active Kubernetes nodes (virtualized or bare metal). This chart does not count "non-participating" nodes in the real-time market, such as suspended nodes.                                                                                                                                                                                                                                                                                 |
| Pod Density             | <p>Average number of pods per node. The 'Optimized' result shows if you can improve density by increasing the number of pods per node.</p> <p>For the total number of pods against the node capacity (maximum pods per node), see the <b>Number of Consumers</b> data in the following charts:</p> <ul style="list-style-type: none"><li>• Nodes (VMs) Optimized Improvements</li><li>• Container Platform Cluster Optimized Improvements</li></ul> |
| Cluster CPU Capacity    | Total CPU capacity for the cluster. The 'Optimized' result indicates how much CPU capacity will result in the optimal number of nodes required to run workloads.                                                                                                                                                                                                                                                                                    |
| Cluster CPU Allocatable | Total amount of cluster CPU available for pod requests. The 'Optimized' result indicates how much of the allocatable CPU capacity will change if you provision or suspend nodes.                                                                                                                                                                                                                                                                    |
| Cluster CPU Request     | Total CPU Request capacity for the cluster.                                                                                                                                                                                                                                                                                                                                                                                                         |
| Cluster CPU Limit       | Total CPU Limit capacity for the cluster.                                                                                                                                                                                                                                                                                                                                                                                                           |

| Row                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Cluster CPU Overcommitment    | <p>(Only for containers with CPU limits) This indicates whether the CPU limits exceed the capacity of the underlying nodes. A value greater than 100% indicates overcommitment. Turbonomic manages cluster resources by actual utilization and limit rightsizing so that you can run more workloads with less risk.</p> <p>Turbonomic only calculates overcommitment in plans. The calculation can be expressed as:</p> <pre data-bbox="873 531 1468 632">Overcommitment = Sum of CPU limits for all containers / Sum of CPU capacity for all nodes</pre>               |
| Cluster Memory Capacity       | Total memory capacity for the cluster. The 'Optimized' result indicates how much memory capacity will result in the optimal number of nodes required to run workloads.                                                                                                                                                                                                                                                                                                                                                                                                  |
| Cluster Memory Allocatable    | Total amount of cluster memory <u>available</u> for pod requests. The 'Optimized' result indicates how much of the allocatable memory capacity will change if you provision or suspend nodes.                                                                                                                                                                                                                                                                                                                                                                           |
| Cluster Memory Request        | Total Memory Request capacity for the cluster.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Cluster Memory Limit          | Total Memory Limit capacity for the cluster.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Cluster Memory Overcommitment | <p>(Only for containers with memory limits) This indicates whether the memory limits exceed the capacity of the underlying nodes. A value greater than 100% indicates overcommitment. Turbonomic manages cluster resources by actual utilization and limit rightsizing so that you can run more workloads with less risk.</p> <p>Turbonomic only calculates overcommitment in plans. The calculation can be expressed as:</p> <pre data-bbox="873 1434 1468 1535">Overcommitment = Sum of memory limits for all containers / Sum of memory capacity for all nodes</pre> |

## Plan Actions

Turbonomic shows separate tabs for **Lift & Shift** and **Optimized** migration actions. You can download the list of actions as a CSV file.



| RESULTS OVERVIEW    |                                             | LIFT & SHIFT ACTIONS (51) | OPTIMIZED ACTIONS (97) |
|---------------------|---------------------------------------------|---------------------------|------------------------|
| RESIZE ^            | Resize Actions (46)                         |                           |                        |
| Workload Co... (46) | <input type="text" value="Type to search"/> |                           |                        |
| MOVE ^              | Workload Controller Name                    | Container Cluster         | Namespace              |
| Container Pods (42) | mediation-awscost                           | Kubernetes-Turbonomic     | turbonomic             |
| START ^             | kafka                                       | Kubernetes-Turbonomic     | turbonomic             |
| Container Pods (6)  | mediation-aws                               | Kubernetes-Turbonomic     | turbonomic             |
| PROVISION ^         | cost                                        | Kubernetes-Turbonomic     | turbonomic             |
| Container Pods (2)  | mediation-gcpsa                             | Kubernetes-Turbonomic     | turbonomic             |

These tabs show the actions that you need to execute to achieve the plan results. For example, you might need to resize limits and requests for containers (via the associated Workload Controllers) to address performance issues. Or, you might need to move pods from one node to another to reduce congestion.

Smarter redistribution and workload rightsizing also drive actions, resulting in the need to provision node(s) based on application demand, or to defragment node resources to enable node suspension.

Turbonomic can recommend the following actions:

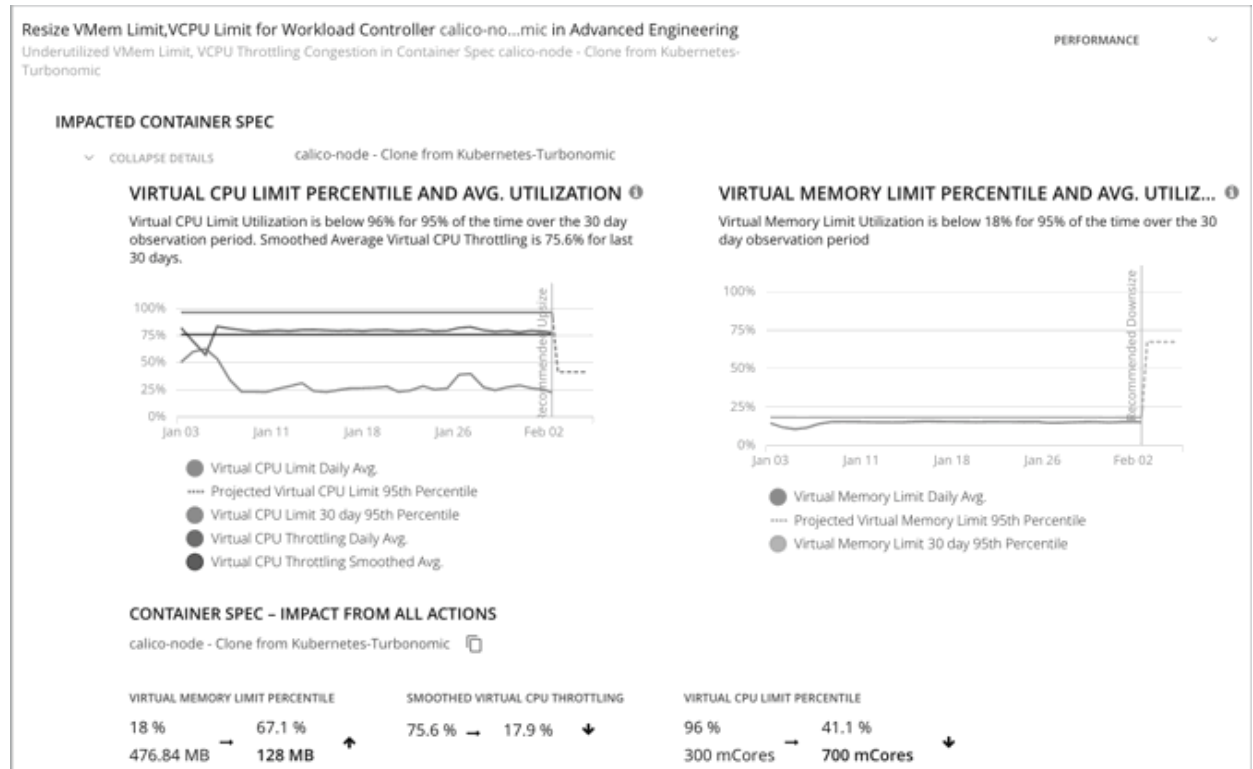
| Entity              | Lift & Shift Actions                                                                                                           | Optimized Actions                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Workload Controller | None                                                                                                                           | Resize container limits and requests<br>See "Workload Controller Resize Actions" below for additional details.                                                                   |
| Namespace           | None                                                                                                                           | Resize quotas (if container resize actions would exceed the namespace quotas)                                                                                                    |
| Pod                 | <ul style="list-style-type: none"> <li>Move</li> <li>Provision (if node provision is required)</li> <li>Reconfigure</li> </ul> | <ul style="list-style-type: none"> <li>Move</li> <li>Provision (if node provision is required)</li> <li>Suspend (if node suspension is required)</li> <li>Reconfigure</li> </ul> |
| Node (VM)           | <ul style="list-style-type: none"> <li>Provision</li> <li>Move on-prem nodes</li> </ul>                                        | <ul style="list-style-type: none"> <li>Provision</li> <li>Suspend</li> <li>Move on-prem nodes</li> </ul>                                                                         |
| Cloud volume        | Scale                                                                                                                          | Scale                                                                                                                                                                            |
| On-prem storage     | <ul style="list-style-type: none"> <li>Provision</li> <li>Suspend</li> </ul>                                                   | <ul style="list-style-type: none"> <li>Provision</li> <li>Suspend</li> </ul>                                                                                                     |

| Entity       | Lift & Shift Actions                                                         | Optimized Actions                                                            |
|--------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| On-prem host | <ul style="list-style-type: none"> <li>Provision</li> <li>Suspend</li> </ul> | <ul style="list-style-type: none"> <li>Provision</li> <li>Suspend</li> </ul> |

Additional information:

- Workload Controller Resize Actions

When you expand an action on a Workload Controller, you will see charts that track VCPU and VMem utilization for the impacted Container Spec. With these charts, you can easily recognize the utilization trends that Turbonomic analyzed to make accurate resize decisions.



For more information about these charts, see [Utilization Charts](#).

Executing several container resize actions can be very disruptive since pods need to restart with each resize. For replicas of the container scale group(s) related to a single Workload Controller, Turbonomic consolidates resize actions into one *merged action* to minimize disruptions. When a merged action has been executed (via the associated Workload Controller), all resizes for all related container specifications will be changed at the same time, and pods will restart once.

- Pod Constraints

A Migrate Container Workloads plan evaluates pod constraints on the source cluster when making placement decisions for pods. The plan enforces taints, tolerations, and nodeSelector specifications if there are matching constraints in the destination cluster. If these constraints cannot be achieved in the destination cluster, the plan ignores them to guarantee placement.

The plan always enforces affinity and anti-affinity constraints, which could result in unplaced pods in the destination cluster. If this happens, the plan generates reconfigure actions for the unplaced pods and shows a notification in the plan results.

⚠ Unplaced Container Pods: Lift&Shift: 2; Optimized: 2 SHOW DETAILS

RESULTS OVERVIEW
LIFT & SHIFT ACTIONS (13)
OPTIMIZED ACTIONS (53)

### Migrate Container Workloads Summary ?

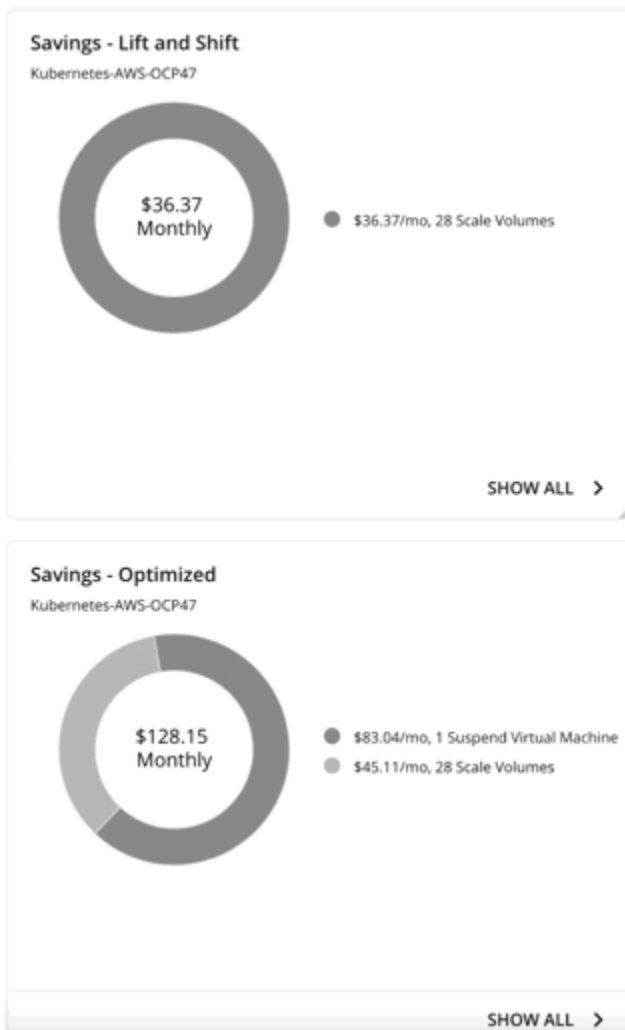
Kubernetes-PT-AKS-Avail

|                      | BEFORE MIGRATION | LIFT & SHIFT | OPTIMIZED |
|----------------------|------------------|--------------|-----------|
| Workload Controllers | 27               | 31           | 31        |
| Container Pods       | 57               | 62           | 57        |

Click **Show Details** to see the list of pods and the reasons for their non-placement. The charts in the plan results do not count these pods.

## Savings

For a destination cluster in the public cloud, Turbonomic shows the savings you would realize if you execute the actions that the plan recommends. The results show separate charts for the **Lift and Shift** and **Optimized** scenarios, so you can compare the impact of actions on your cloud expenses.



For example, in both scenarios, the plan might recommend scaling volumes to new tiers to address performance issues. If these new tiers happen to be more cost-effective than the current tiers, then the actions are treated as cost-saving measures. For the optimized scenario, the plan might also recommend suspending certain nodes to increase infrastructure efficiency, which could introduce additional savings.

Note that application performance and infrastructure efficiency are the drivers of these actions, *not* cost optimization. Cost information is included to help you track your cloud expenses.

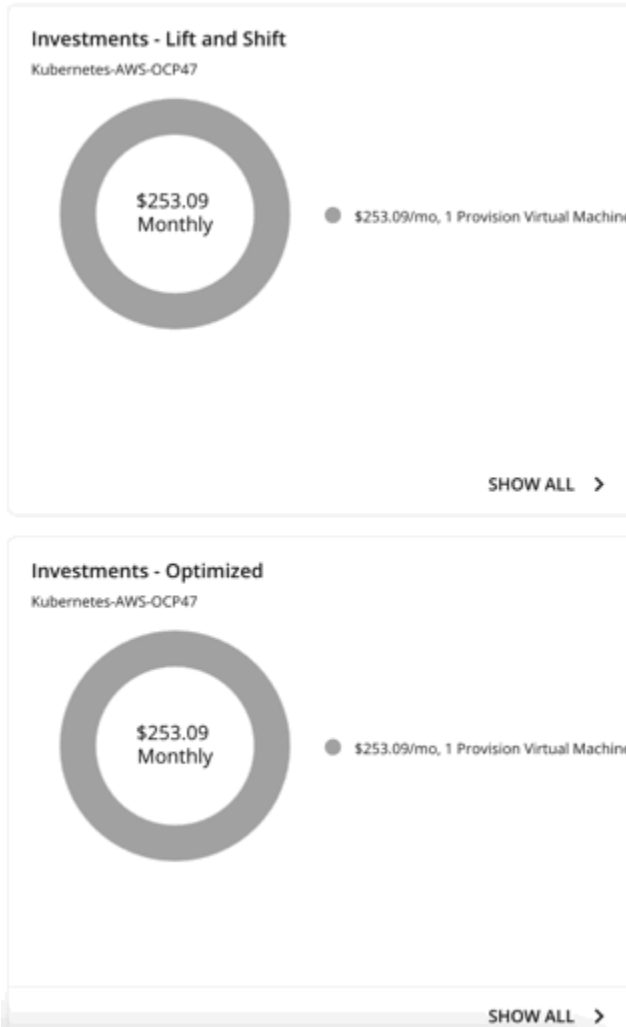
The charts show total monthly savings. Click **Show All** to view the actions with cost savings.

**Note:**

An empty chart indicates that no savings will be realized after you execute the recommended actions.

## Investments

For a destination cluster in the public cloud, Turbonomic shows the costs you would incur if you execute the actions that the plan recommends. The results show separate charts for the **Optimized** and **Lift and Shift** scenarios, so you can compare the impact of actions on your cloud expenses.



For example, if some applications risk losing performance, the plan might recommend provisioning nodes to increase capacity. The charts show how node provision actions translate to an increase in expenditure. You can use cost information in the charts as you seek approval to execute these actions.

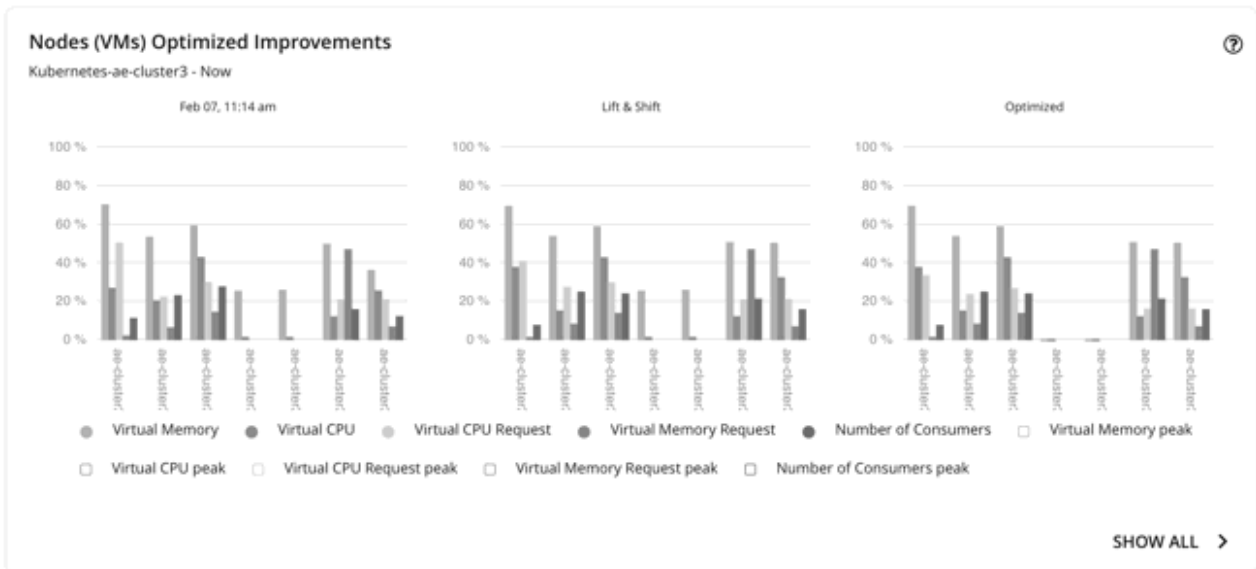
The charts show total monthly investments. Click **Show All** to view the actions that require investments.

**Note:**

An empty chart indicates that no investments are required to execute the recommended actions.

## Nodes (VMs) Optimized Improvements

This chart compares the following before and after the plan:

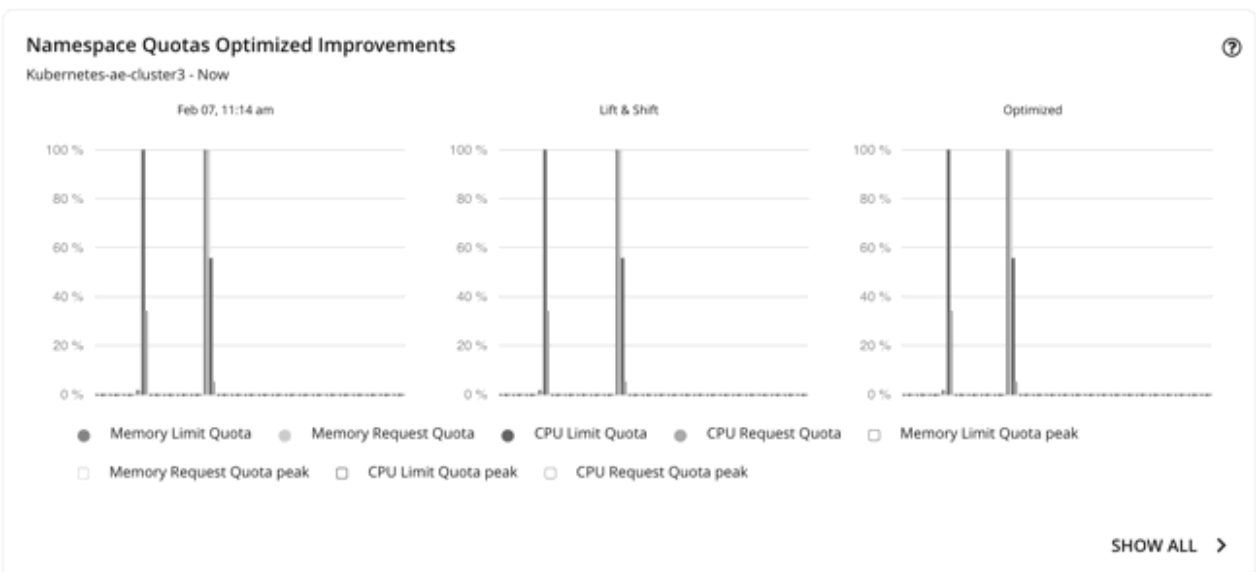


- Utilization of the following resources for all nodes:
  - vMem
  - vCPU
  - vMem Request
  - vCPU Request
- Number of pods consuming resources against the maximum pod capacity for all the nodes

Use this chart to identify nodes that are candidates for suspension, or resources with unusual utilization. You can also use this chart to drill down to a specific resource. Click **Show All** for more details.

### Namespace Quotas Optimized Improvements

This chart shows pod utilization of resource quotas defined in namespaces. Resource quotas include:

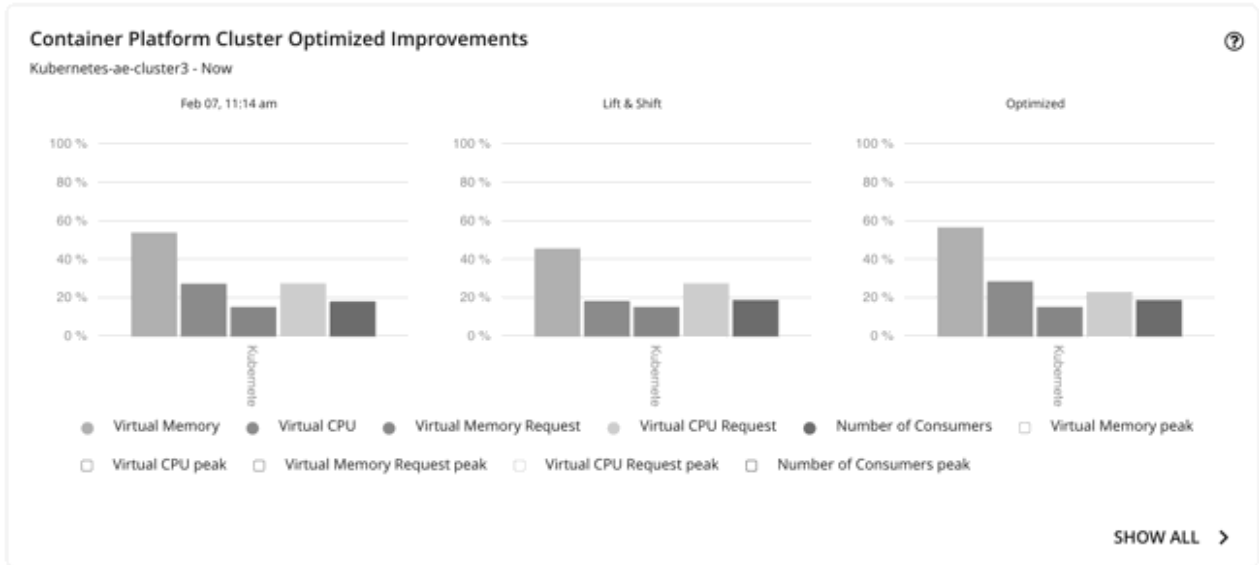


- CPU Limit Quota
- Memory Limit Quota
- CPU Request Quota
- Memory Request Quota

The chart highlights utilization improvements as a result of namespace resizing. For namespaces without defined quotas, utilization is 0 (zero).

## Container Platform Cluster Optimized Improvements

This chart shows the following, assuming you execute all actions in the plan:



- 
- Changes to the utilization of cluster resources
- Overcommitment values

## Optimized Improvements for Hosts and Storage

Use these charts if you ran the plan on an on-prem cluster. These charts show how the utilization of resources would change assuming you accept all of the actions that the plan recommends.

## Downloading Plan Results


You can download the plan results shown in individual charts. Click the **Show All** button for a chart, and then the download button at the top-right section of the Details page.

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.

### Note:

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Optimize Cloud Plan

Run the Optimize Cloud plan to see how you can maximize savings while still assuring performance for your applications and workloads. This plan identifies ways to optimize your costs by choosing the best templates (most adequate compute resources), regions, accounts, or resource groups to host your workloads. The plan also identifies workloads that can change over to discounted pricing, and it compares your current costs to the costs you would get after executing the plan recommendations.

The screenshot shows the 'Optimize Cloud 56' interface. The 'SCOPE' is set to 'Prod'. A 'RUN AGAIN' button is visible. The 'CONFIGURATION' section on the left includes settings for 'RI Profile', 'Buy RIs', 'AWS: Offering Class', 'AWS: Term', 'AWS: Payment', 'RI Inventory', and 'Virtual Machine Action Settings'. The 'RESULTS OVERVIEW' section displays a 'Cloud Cost Comparison' table.

|                                         | CURRENT      | OPTIMIZED    | DIFFERENCE   | %         |
|-----------------------------------------|--------------|--------------|--------------|-----------|
| Workloads with performance risks        | 0 Out Of 125 | 0 Out Of 125 | 0            | -         |
| Workloads with efficiency opportunities | 0 Out Of 125 | 0 Out Of 125 | 0            | -         |
| Workloads out of compliance             | 0 Out Of 125 | 0 Out Of 125 | 0            | -         |
| RI Coverage                             | 21 %         | 92 %         | ▲ 338.1 %    |           |
| RI Utilization                          | 49 %         | 16 %         | ▼ 67.3 %     |           |
| On-Demand Compute Cost                  | \$10,330 /mo | \$2,961 /mo  | -\$7,369 /mo | ▼ 71.3 %  |
| Reserved Compute Cost                   | \$2,123 /mo  | \$6,802 /mo  | \$4,679 /mo  | ▲ 220.4 % |
| On-Demand Database Cost                 | \$1,634 /mo  | \$1,634 /mo  | \$0 /mo      | 0 %       |
| Storage Cost                            | \$3,442 /mo  | \$3,442 /mo  | \$0 /mo      | 0 %       |
| Total Cost                              | \$17,529 /mo | \$14,839 /mo | -\$2,690 /mo | ▼ 15.3 %  |

### Configuring an Optimize Cloud Plan

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

#### 1. Scope

You can scope by:

The 'Select Scope to Optimize' dialog box shows a list of scope options. 'CLOUD' is selected, and 'All Providers' is chosen under it. Other options include 'Accounts (7)', 'Billing Families (3)', 'Cloud Providers (2)', 'Resource Groups (35)', and 'Regions (43)'. The 'All Providers' option is currently selected with a checkmark.

- Accounts

Choose an AWS account or Azure subscription for the plan's scope. If you choose an Account for the scope, then the plan will not recommend discount purchases. To optimize discount purchases for a limited scope, choose a Billing Family.

- Billing Families

Include discount purchases in the planning for a scope that is limited to a single billing family. The plan calculates discount purchases through the billing family's master account.

- Cloud Providers

See how you can optimize all your AWS or Azure workloads.

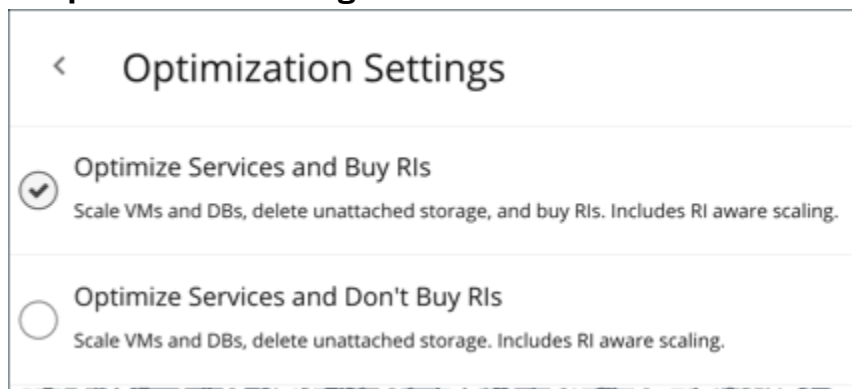
- Resource Groups

Turbonomic discovers Azure resource groups. You can select one or more resource groups for the plan scope.

- Regions

Focus the plan on a provider's region.

## 2. Optimization Settings



Choose from the given optimization options. Note that if you set a plan's scope to a resource group, Turbonomic will optimize services without recommending discount purchases.

If your goal is to purchase discounts for VMs at their current sizes, use the Buy VM Reservations plan type. For details, see [Buy VM Reservations Plan](#).

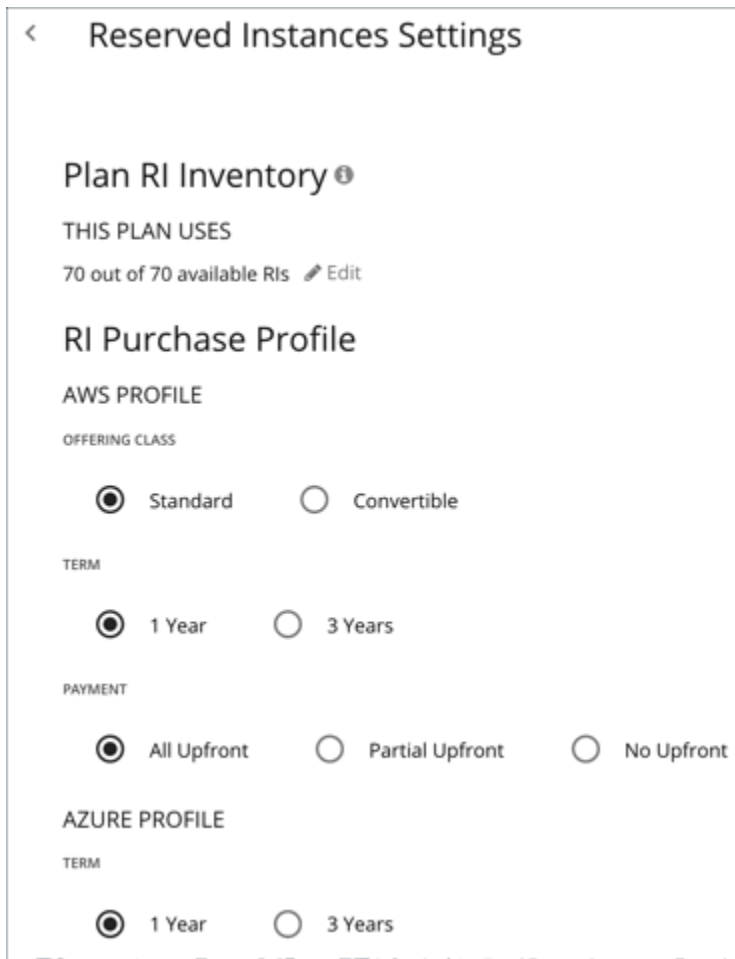
### Note:

If you turn on the **Disable All Actions** setting in the global default policy and then run an Optimize Cloud plan with VM scaling and discount purchases enabled, the plan results show inaccurate discount recommendations.

Turn off **Disable All Actions** to resolve this issue. Be aware that after you turn off this setting, it will take Turbonomic a week to reflect accurate results in Optimize Cloud plans.

## 3. Reserved Instances Settings





For **Plan RI Inventory**, the discounts for the current scope are selected by default. Click **Edit** to make changes.

For **RI Purchase Profile**, the settings that you have set up for real-time analysis are selected by default. You can change the settings to see how they affect costs.

- Offering Class

For AWS environments, choose the offering class that corresponds to the RI types that you typically use in your environment.

- Term

For AWS and Azure environments, choose the payment terms you contract for your discounts. TERM can be one of **1 Year** or **3 Year**. Typically, longer term payment plans cost less per year.

- Payment

The payment option that you prefer for your AWS RIs:

- All Upfront – You make full payment at the start of the RI term.
- Partial Upfront – You make a portion of the payment at the start of the term, with the remain cost paid at an hourly rate.
- No Upfront – You pay for the RIs at an hourly rate, for the duration of the term.

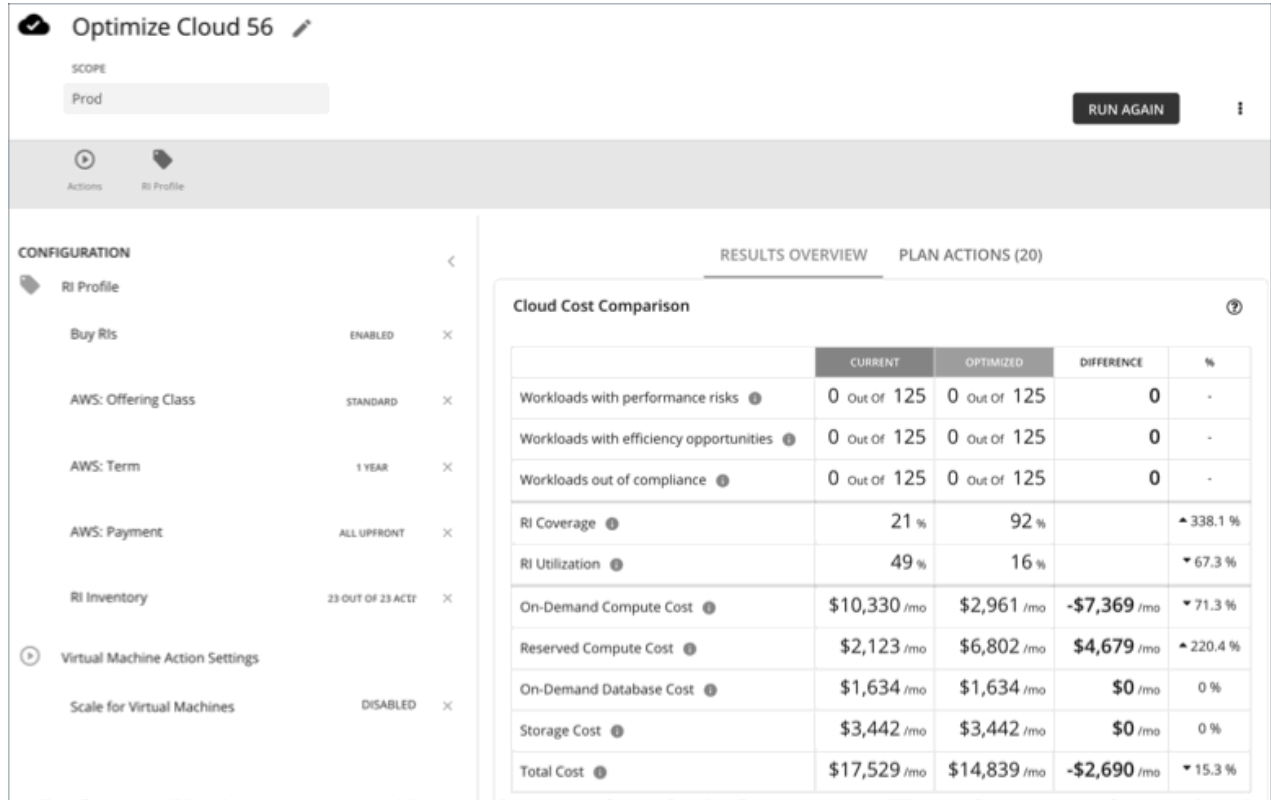
### ***Working With Optimize Cloud Plan Results***

After the Optimize Cloud plan runs, you can view the results to see how you can maximize savings or make other improvements to your cloud environment.

The plan results:

- Compare current to optimized costs, including on-demand compute, discounted compute, on-demand database, and storage costs
- Compare current and optimized breakdowns of templates used
- Compare breakdowns of storage tiers in use
- Project the discount coverage (how many workloads are covered by discounted pricing) and utilization (percentage of discounts that are active)
- Show the cost benefit of moving workloads from on-demand to discounted pricing

## Viewing the Results



The plan results include the following charts:

- **Cloud Cost Comparison**

This chart highlights any difference in cost as a result of optimization. For example, undersized VMs risk losing performance and should therefore scale up. This could contribute to an increase in cost. On the other hand, oversized VMs can scale down to less expensive instances, so cost should go down. The values under the **%** column indicate the percentage of VMs that are affected by optimization cost calculations.

Turbonomic can also recommend that you purchase discounts to reduce costs. The analysis looks at workload history to identify workloads that can move from on-demand to discounted pricing. This considers the count of workloads in a family, plus their hours of active-state condition, to arrive at the discounts you should purchase. Since discounted costs are incurred at the account level, the Cloud Cost Comparison chart will present discounted costs or charges when you scope to an account or group of accounts (including a billing family).

For AWS clouds, Turbonomic can get the information it needs to display license costs for database instances. For Azure clouds, Turbonomic does not display database license costs because Azure does not make that information available.

- **Workload Mapping**

This chart shows the types of tiers you currently use, compared to the tiers the plan recommends, including how many of each type, plus the costs for each.

To see a detailed breakdown of the template costs, click **SHOW CHANGES** at the bottom of the chart.

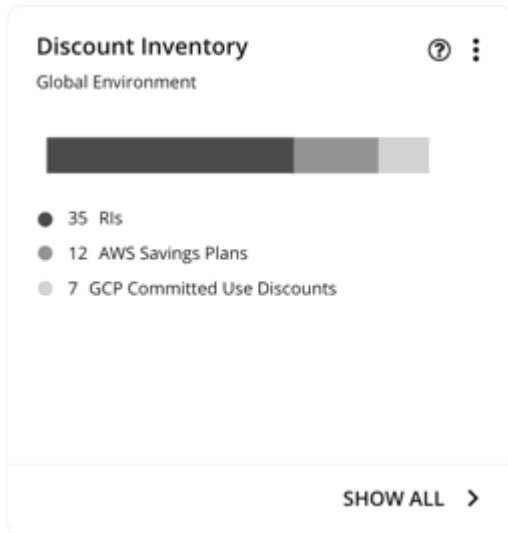
- **Volume Tier Summary**

This chart shows the current distribution of volumes that support your workloads, and the optimized distribution if you execute the actions that the plan recommends.

The difference in the result reflects the number of unattached volumes. To see a list of unattached volumes, click **Show changes** at the bottom of the chart.

- **Discount Inventory**

This chart lists the cloud provider discounts discovered in your environment. For a tabular listing, click **Show All** at the bottom of the chart. In the tabular listing, you can see if a discount expired before the specified purchase date.



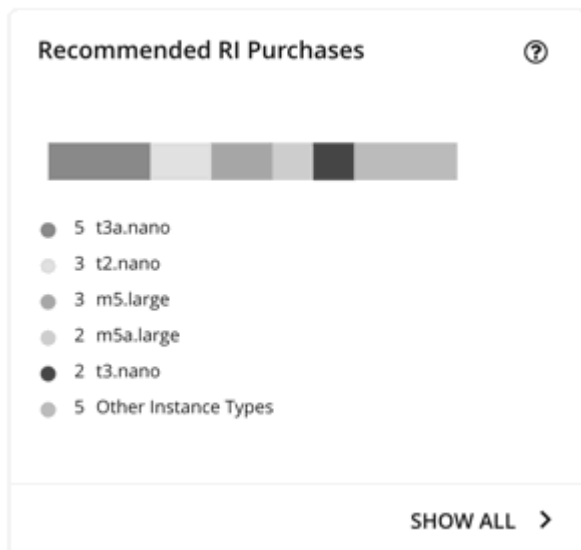
- **Recommended RI Purchases**

Turbonomic can recommend purchasing instance types at a discounted rate to help you increase the percentage of VMs covered by discounted pricing and reduce on-demand costs. This chart shows your pending purchases. Download the list of purchases and then send it your cloud provider or representative to initiate the purchase process.

**Note:**

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

Currently, Turbonomic can recommend purchase actions for AWS and Azure. Purchase actions for GCP will be introduced in a future release.



Click **Show All** to see a table with details for each discount.

The table shows the properties, up-front cost, and break-even period for each discount. The break-even period is the time at which savings will exceed the up-front cost, rounded to the month. The Cost Impact column indicates the monthly savings you would realize when you buy a specific discount.

When you choose one or more check boxes, the total count, up-front cost, and savings appear at the top.

## Viewing Plan Actions

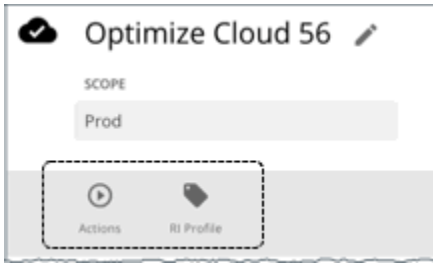
Click the **Plan Actions** tab on top of the page to view a list of actions that you need to execute to achieve the plan results.

|                        |  | RESULTS OVERVIEW          | PLAN ACTIONS (274) |                   |                   |                   |                       |                    |                 |               |
|------------------------|--|---------------------------|--------------------|-------------------|-------------------|-------------------|-----------------------|--------------------|-----------------|---------------|
| <b>DELETE</b> ^        |  | <b>Scale Actions (28)</b> |                    | TOTAL SAVINGS     | TOTAL INVESTMENTS |                   |                       |                    |                 |               |
| Volumes (131)          |  |                           |                    | \$3,111.75/mo     | \$242.80/mo       |                   |                       |                    |                 |               |
| <b>SCALE</b> ^         |  | Type to search            |                    |                   |                   |                   |                       |                    |                 | ADD FILTER    |
| Volumes (79)           |  | Virtual Machine Name      | Instance Type      | Discount Coverage | On-Demand Cost    | New Instance Type | New Discount Coverage | New On-Demand Cost | Action Category | Cost Impact   |
| Virtual Machines (28)  |  | eks-cluster-eks-cls       | m5a.4xlarge        | 0%                | \$0.684/h         | r5a.2xlarge       | 75%                   | \$0.112/h          | SAVINGS         | ↓ \$417.49/mo |
| Database Servers (7)   |  | ocp47demo-2v5jc           | m5a.4xlarge        | 0%                | \$0.684/h         | r5a.2xlarge       | 75%                   | \$0.112/h          | SAVINGS         | ↓ \$417.49/mo |
| <b>BUY</b> ^           |  | cn-latest-lzdjn-wci       | m6i.2xlarge        | 0%                | \$0.382/h         | r5a.xlarge        | 100%                  | \$0.00/h           | SAVINGS         | ↓ \$278.81/mo |
| Reserved Instance (15) |  | cn-latest-lzdjn-wci       | m6i.2xlarge        | 0%                | \$0.382/h         | r5.xlarge         | 50%                   | \$0.125/h          | SAVINGS         | ↓ \$187.33/mo |
|                        |  | pl-ocp-apm-gr2qp          | r5a.xlarge         | 0%                | \$0.247/h         | r5.xlarge         | 100%                  | \$0.00/h           | PERFORMANCE     | ↓ \$180.07/mo |

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.




- Actions

Use this to enable or disable automatic Scale actions for the virtual machines in the plan.

- RI Settings

See [Reserved Instance Settings](#).

**Note:**

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Migrate to Cloud Plan

|                                                | LIFT & SHIFT         | OPTIMIZED           | DIFFERENCE           | %               |
|------------------------------------------------|----------------------|---------------------|----------------------|-----------------|
| Virtual Machines with performance risks        | 99 Out Of 684        | 0 Out Of 687        | 99                   | -               |
| Virtual Machines with efficiency opportunities | 588 Out Of 684       | 0 Out Of 687        | 588                  | -               |
| On-Demand Compute Cost                         | \$113,460 /mo        | \$23,214 /mo        | -\$90,246 /mo        | ▼ 79.5 %        |
| Reserved Compute Cost                          | \$9,816 /mo          | \$19,229 /mo        | \$9,413 /mo          | ▲ 95.9 %        |
| Storage Cost                                   | \$32,183 /mo         | \$40,397 /mo        | \$8,214 /mo          | ▲ 25.5 %        |
| <b>Total Cost</b>                              | <b>\$155,459 /mo</b> | <b>\$82,840 /mo</b> | <b>-\$72,619 /mo</b> | <b>▼ 46.7 %</b> |

A Migrate to Cloud plan simulates migration of on-prem VMs to the cloud, or migration of VMs from one cloud provider to another. This plan focuses on optimizing performance and costs by choosing the most suitable cloud resources for your VMs and the volumes they use. To further optimize your costs, the plan can recommend moving workloads from on-demand to discounted pricing, and purchasing more discounts.

The plan calculates costs according to the billing and price adjustments that you have negotiated with your cloud provider. Costs include compute, service (such as IP services), and license costs. The plan also calculates discount purchases for VMs that can benefit from discounted pricing.

**Note:**

If your instance of Turbonomic is inoperative for a period of time, that can affect the cost calculations. To calculate costs for a VM that it will migrate to the cloud, Turbonomic considers the VM's history. For example, if the VM has been stable for 16 of the last 21 days, then Turbonomic will plan for that VM to

use a discount. In this way, the plan calculates the best cost for the migration. However, if Turbonomic is inoperative for any time, that can impact the historical data such that the plan will *not* recognize a VM as stable, even though it is.

Points to consider:

- AWS includes EC2 Spot Instances that offer steep discounts. A plan that migrates from AWS to Azure will not migrate VMs that run on Spot Instances.
- Do not use this plan type to migrate within the same cloud provider (for example, moving VMs from one Azure subscription to another) as a way to test the effect on pricing. The results from such a plan would not be reliable.
- For migrations within your on-prem environment, use the *Virtual Machine Migration* plan type.
- Before migrating, consider turning on a setting in the default global policy that enables metrics collection for on-prem volumes attached to VMs. This allows Turbonomic to make more accurate placement decisions for the VMs and volumes you are migrating. For details, see [Enable Analysis of On-prem Volumes](#).

The plan results show:

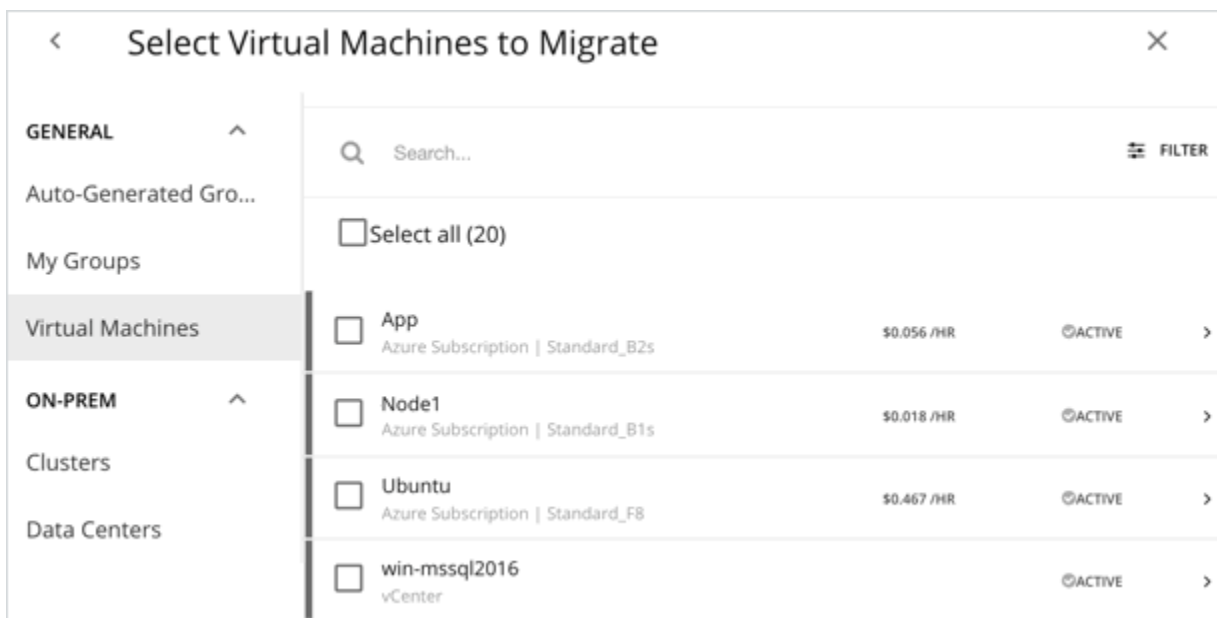
- Projected costs
- Actions to execute your migration and optimize costs and performance
- Optimal cloud instances to use, combining efficient purchase of resources with assured application performance
- The cost benefit of moving workloads from on-demand to discounted pricing
- Discounts you should purchase

## Configuring a Migrate to Cloud Plan

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

### 1. Scope

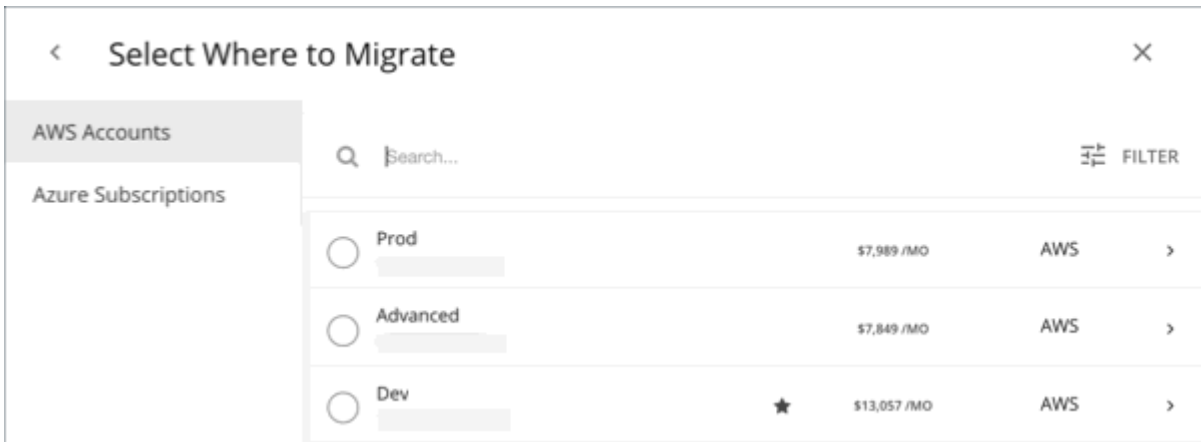
Select the VMs that you want to migrate. You can select VM groups and/or individual VMs.



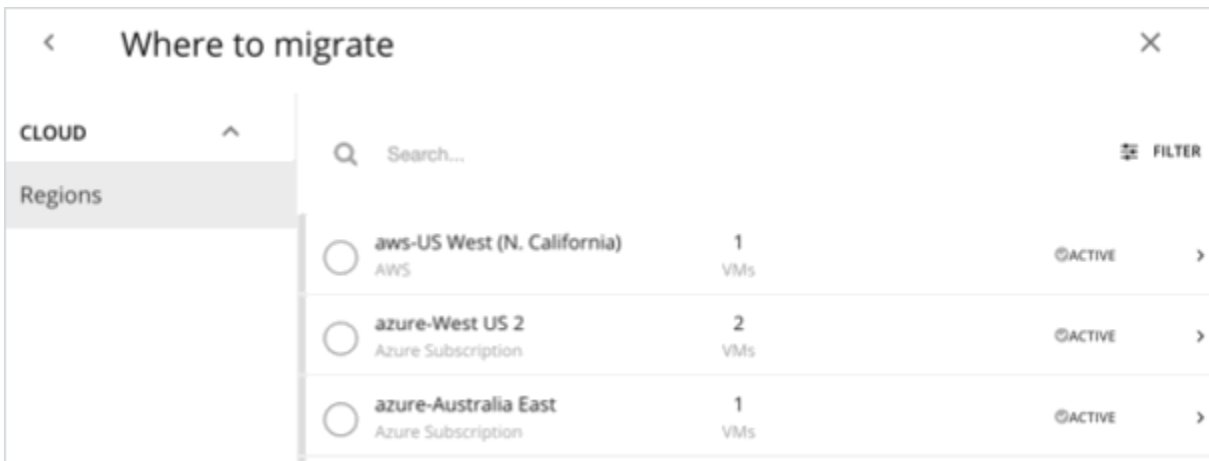
If you select an Auto Scaling Group, Turbonomic simulates migrating the VMs individually, and not as a group.

## 2. Where to Migrate

Choose a billing account (AWS account or Azure subscription).



Choose a region. Turbonomic shows all the regions that you can access from your target cloud accounts.



By default, Turbonomic considers all instance types in the selected region when making placement decisions for the scoped VMs and the volumes they use. However, you may have set up constraints in policies that limit migration to certain instance types. If there are VMs and volumes in your scope that are affected by those policies, Turbonomic will only consider the instance types defined in the policies.

## 3. Licensing (OS Migration Profile)

Select an OS Profile for this migration.

< Licensing

**Include OS cost**  
The plan includes OS cost per VM as part of the migration.

**BYOL ("Bring your own license")**  
The plan does not include OS cost as part of the migration. Use this option if you have purchased Azure Hybrid Benefit.

**Custom OS**  
The plan includes OS cost based on your selections.

| SOURCE OS | TARGET OS | BYOL                                |
|-----------|-----------|-------------------------------------|
| Linux     | Linux     | <input checked="" type="checkbox"/> |
| RHEL      | RHEL      | <input checked="" type="checkbox"/> |
| SLES      | SLES      | <input checked="" type="checkbox"/> |
| Windows   | Windows   | <input checked="" type="checkbox"/> |

On the cloud, instances usually include an OS platform to run processes on the VM. As you migrate VMs to the cloud, you can specify the OS you prefer to run. You can keep the same OS that the original VM has, or map it to a different OS.

- Include OS cost

As Turbonomic calculates placement for the migrated workloads, it will include costs for instances that provide the same OS that the VM already has.

- BYOL (Bring your own license)

This is the same as the **Include OS cost** option, except the plan does not include OS licensing costs in any of the cost calculations for on-cloud placement.

- Custom OS

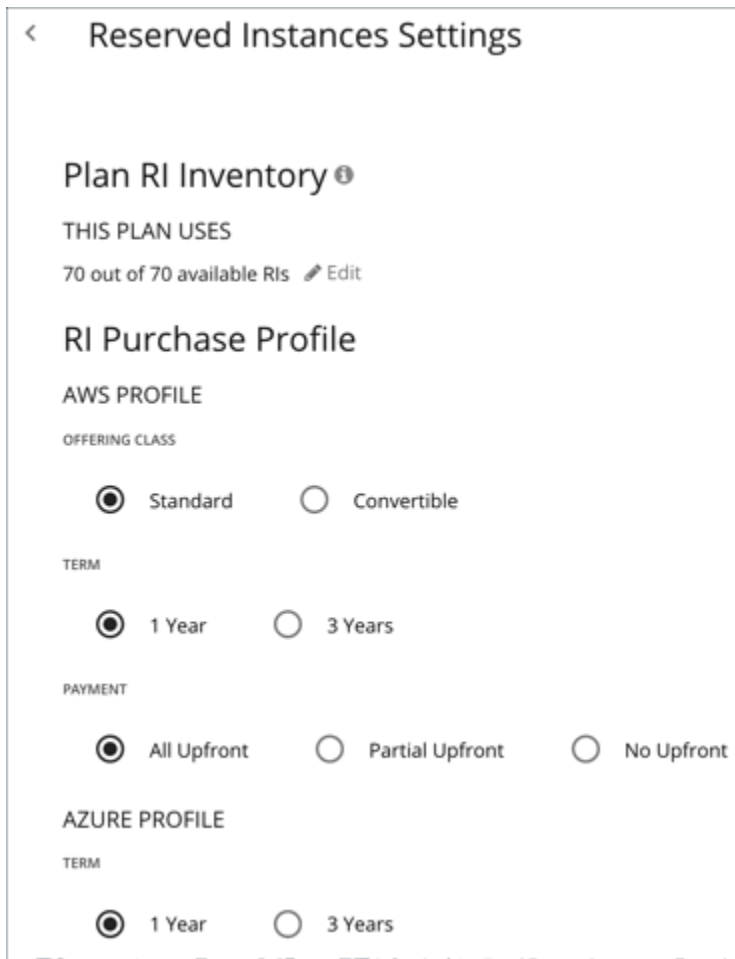
For each of the listed OS types, map the migrated VM to the OS you choose. The OS types are:

- Linux – Any open source distribution of Linux. For the migration, Turbonomic will choose instances that provide the Linux platform that the cloud service provider delivers as a free platform. Note that this is always BYOL, because it assumes a free OS license.
- RHEL (Red Hat Enterprise Linux)
- SLES (SUSE Linux Enterprise Server)
- Windows

If you enable **BYOL** for RHEL, SLES, or Windows, Turbonomic assumes that you are paying for the OS license, and will not include the license cost in the plan results. If you do not enable **BYOL**, Turbonomic gets the license cost from the service provider and includes that cost in the plan results.

## 4. Reserved Instances Settings





For **Plan RI Inventory**, the discounts for the current scope are selected by default. Click **Edit** to make changes.

For **RI Purchase Profile**, the settings that you have set up for real-time analysis are selected by default. You can change the settings to see how they affect costs.

- Offering Class

For AWS environments, choose the offering class that corresponds to the RI types that you typically use in your environment.

- Term

For AWS and Azure environments, choose the payment terms you contract for your discounts. TERM can be one of **1 Year** or **3 Year**. Typically, longer term payment plans cost less per year.

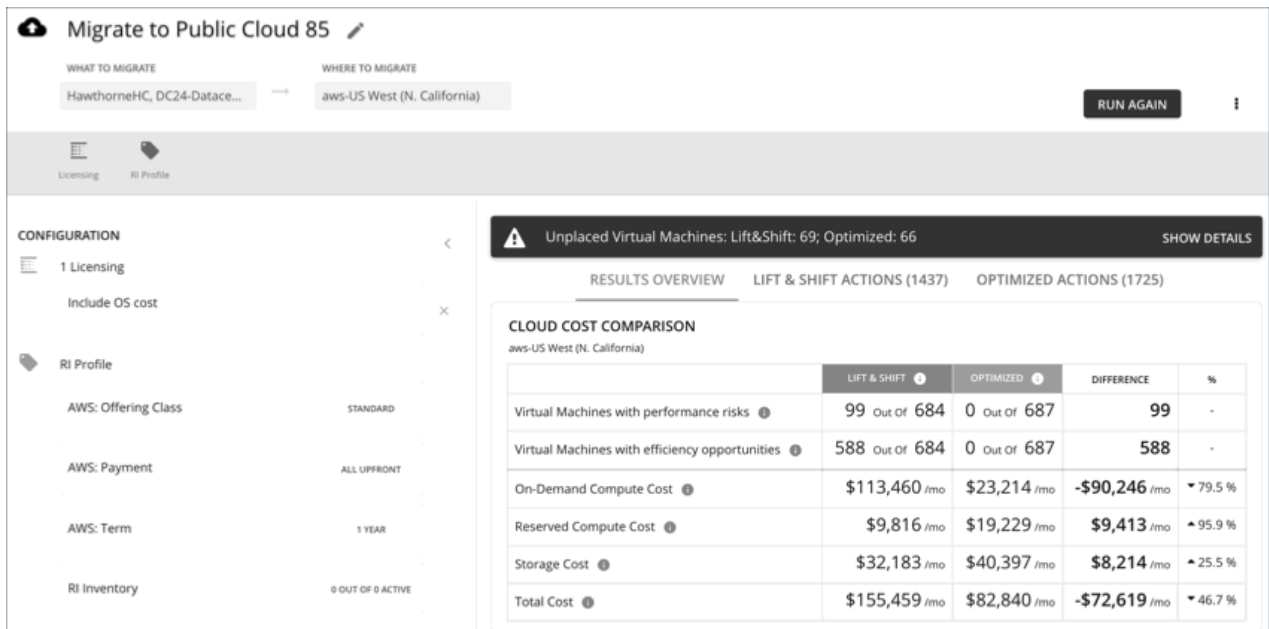
- Payment

The payment option that you prefer for your AWS RIs:

- All Upfront – You make full payment at the start of the RI term.
- Partial Upfront – You make a portion of the payment at the start of the term, with the remain cost paid at an hourly rate.
- No Upfront – You pay for the RIs at an hourly rate, for the duration of the term.

### ***Working With Migrate to Cloud Plan Results***

The Migrate to Cloud plan results show the cloud resources and costs for the VMs you plan to migrate, and the actions required for migration.



Turbonomic shows results for two migration scenarios:

- **Lift & Shift**

Lift & Shift migrates your VMs to cloud instances that match their current resource allocations.

- **Optimized**

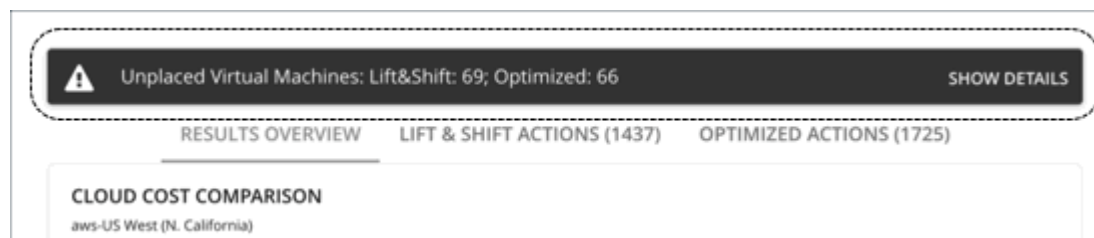
As Turbonomic runs the plan, it looks for opportunities to optimize cost and performance. For example, it might discover overprovisioned VMs after analyzing the historical utilization of VM resources. If you were to migrate such VMs to instances that match their current allocations, then you would spend more than necessary. For an optimized migration, Turbonomic can recommend migrating to less expensive instances while still assuring performance, and then show the resulting savings. In addition, when you examine the actions for an optimized migration, you will see charts that plot the historical utilization data used in the analysis.

## Results Overview

The Results Overview section shows the following:

- **Unplaced VMs**

If the plan's scope includes VMs that cannot be migrated, the results include a notification indicating the number of VMs. Click **Show Details** to see the list of VMs and the reasons for their non-placement.



The charts in the plan results do not count these VMs.

Turbonomic displays adjusted CPU values for unplaced VMs. These values are the actual metrics used in analysis and are calculated using benchmark data. CPU values shown in other places (such as the Capacity and Usage chart) are unadjusted values obtained from targets.

- **Cloud Cost Comparison Chart**

This chart highlights any difference in cost as a result of optimization. For example, undersized VMs risk losing performance and should therefore scale up. This could contribute to an increase in cost. On the other hand, oversized VMs can scale down to less expensive instances, so cost should go down. The values under the **%** column indicate the percentage of VMs that are affected by optimization cost calculations.

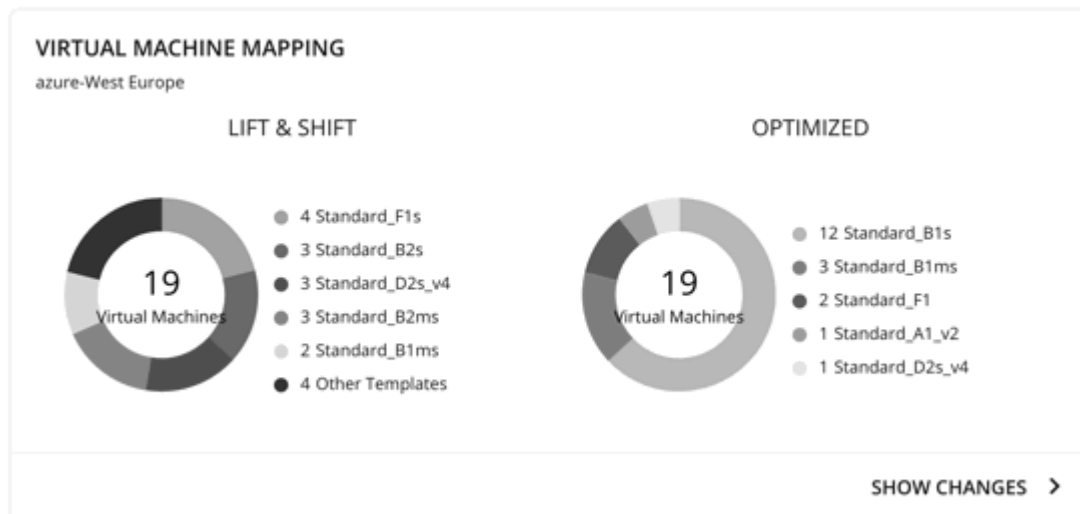
| CLOUD COST COMPARISON                          |                      |                     |                      |                 |
|------------------------------------------------|----------------------|---------------------|----------------------|-----------------|
| aws-US West (N. California)                    |                      |                     |                      |                 |
|                                                | LIFT & SHIFT         | OPTIMIZED           | DIFFERENCE           | %               |
| Virtual Machines with performance risks        | 99 Out Of 684        | 0 Out Of 687        | 99                   | -               |
| Virtual Machines with efficiency opportunities | 588 Out Of 684       | 0 Out Of 687        | 588                  | -               |
| On-Demand Compute Cost                         | \$113,460 /mo        | \$23,214 /mo        | -\$90,246 /mo        | ▼ 79.5 %        |
| Reserved Compute Cost                          | \$9,816 /mo          | \$19,229 /mo        | \$9,413 /mo          | ▲ 95.9 %        |
| Storage Cost                                   | \$32,183 /mo         | \$40,397 /mo        | \$8,214 /mo          | ▲ 25.5 %        |
| <b>Total Cost</b>                              | <b>\$155,459 /mo</b> | <b>\$82,840 /mo</b> | <b>-\$72,619 /mo</b> | <b>▼ 46.7 %</b> |

**Note:**

For Azure, the results do not include the license cost for the migrated VMs.

- **Virtual Machine Mapping Chart**

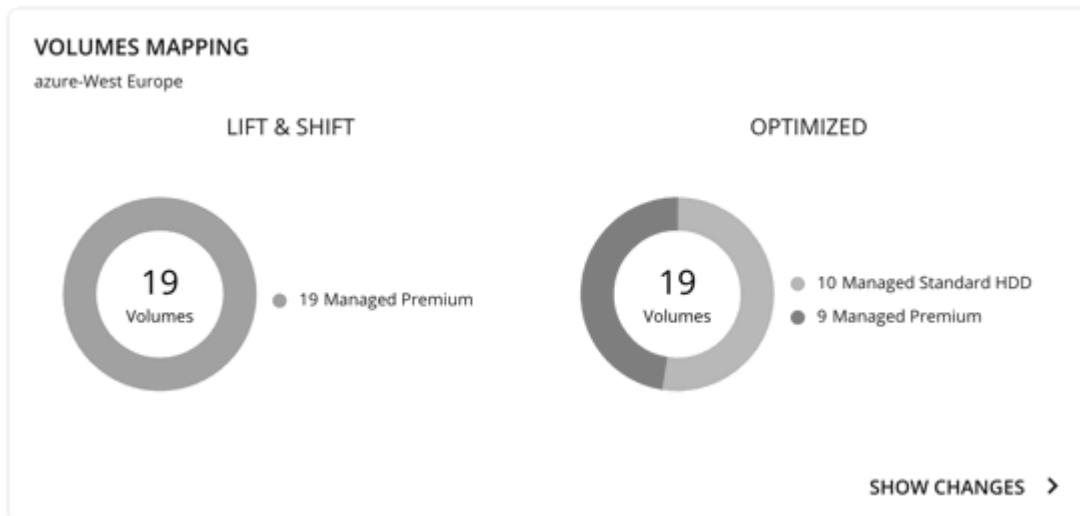
This chart gives a breakdown of the instance types that the plan recommends for the migration, including how many of each is needed.



Click **Show Changes** to see a table with details for each VM in the plan. The table maps VMs to instance types. It also shows the properties and monthly cost for each instance type, and indicates whether Turbonomic recommends buying discounts. Under the **Actions** column, click **Details** to compare Lift & Shift and Optimized actions.

- **Volume Tier Summary Chart**

This chart gives a breakdown of the volume types that the plan recommends for the migration, including how many of each is needed.



Click **Show Changes** to see a table with details for each volume in the plan. The table maps the volumes you plan to migrate to the volume types that Turbonomic recommends. It also shows the properties and monthly cost for each volume type. Under the **Actions** column, click **Details** to compare Lift & Shift and Optimized actions.

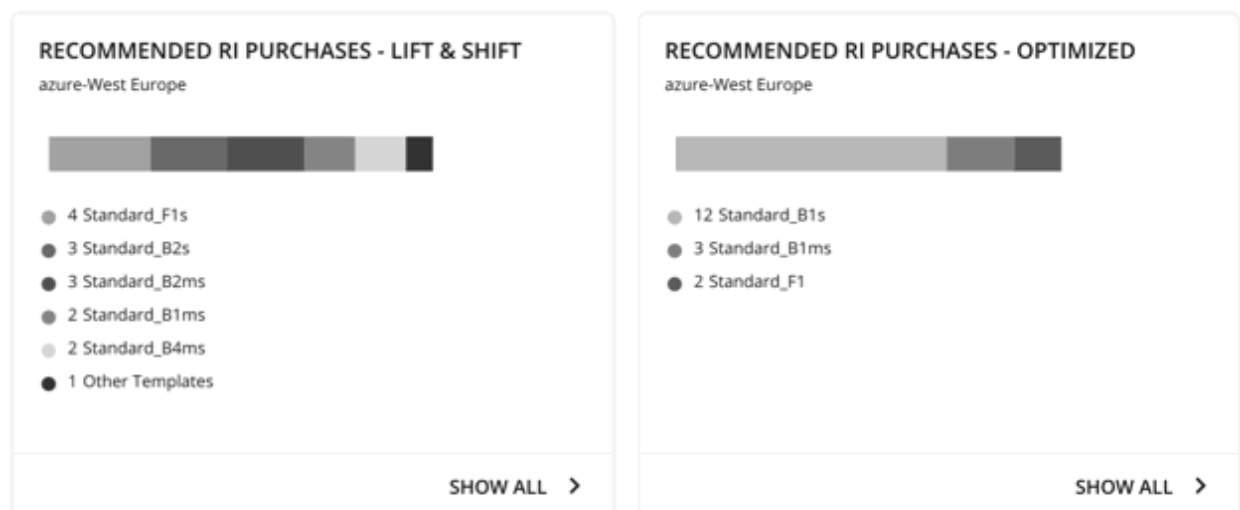
• **Recommended RI Purchases Charts**

Turbonomic can recommend purchasing instance types at a discounted rate to help you increase the percentage of VMs covered by discounted pricing and reduce on-demand costs. This chart shows your pending purchases. Download the list of purchases and then send it your cloud provider or representative to initiate the purchase process.

**Note:**

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

Currently, Turbonomic can recommend purchase actions for AWS and Azure. Purchase actions for GCP will be introduced in a future release.



To identify VMs that are good candidates for discounted pricing, Turbonomic analysis considers the history of a VM (by default, the last 21 days), and it looks for:

- Activity
  - If the VM's VCPU utilization percentile is 20% or higher, then Turbonomic considers it an active VM.
- Stability

If there have been no start, stop, or resize actions for the VM for 16 of the last 21 days, then Turbonomic considers it stable.

If the current discount inventory cannot support the VM, or if supporting it would exceed your desired coverage, then Turbonomic can recommend purchasing additional discounts.

Click **Show All** to see a table with details for each discount.

The table shows the properties, up-front cost, and break-even period for each discount. The break-even period is the time at which savings will exceed the up-front cost, rounded to the month. The Cost Impact column indicates the monthly savings you would realize when you buy a specific discount.

When you choose one or more check boxes, the total count, up-front cost, and savings appear at the top.

Click **Details** under the **Actions** column to compare Lift & Shift and Optimized actions.

**Note:**

The plan assumes that a discount will always be less expensive than its on-demand counterpart. However, this is not always the case. There might be billing details from service providers that could lead to recommendations to move to a discounted instance type that is more expensive than running on demand.

## Plan Actions

Turbonomic shows separate tabs for **Lift & Shift** and **Optimized** migration actions. You can download the list of actions as a CSV file.

| RESULTS OVERVIEW     |                      | LIFT & SHIFT ACTIONS (2) |               | OPTIMIZED ACTIONS (2) |                     |                 |          |
|----------------------|----------------------|--------------------------|---------------|-----------------------|---------------------|-----------------|----------|
| MOVE                 | Move Actions (1)     |                          |               |                       |                     | Settings        | Download |
| Volumes (1)          | Type to search       |                          |               |                       |                     | ADD FILTER      |          |
| Virtual Machines (1) | Virtual Machine Name | Type                     | From          | To                    | Risk                | Action Category | Action   |
|                      | pysavingAz1          | Computer Tier            | Standard_B1ms | m5.large              | Optimized migration | PERFORMANCE     | DETAILS  |

For *Optimized* migrations, when you expand an action on a VM, you will see charts that track VCPU and VMem utilization for that VM. With these charts, you can easily recognize the utilization trends that Turbonomic analyzed to determine the most efficient instance for the VM.



For more information about these charts, see [Utilization Charts](#).

## Uploading Plan Results to Azure Migrate

Turbonomic can upload the plan results and additional plan information to the Azure Migrate portal as part of your migration process. This feature is only available for plans that simulate on-prem VM migration to an Azure region.

Uploaded information includes:

- Basic information for the on-prem VMs, including OS Name and Machine Name
- Target Azure region, VM size, and storage type

### Note:

Azure Migrate does not support automatic selection of OS Disk or manual selection of Ultra Storage disk tiers as part of a migration plan.

- Discount recommendations
- OS license recommendations (based on the licensing option that you selected for the plan)

### Note:

The Azure Migrate portal displays standardized information provided by third-party migration assessment solutions, including Turbonomic. Microsoft might not support displaying some information unique to Turbonomic.

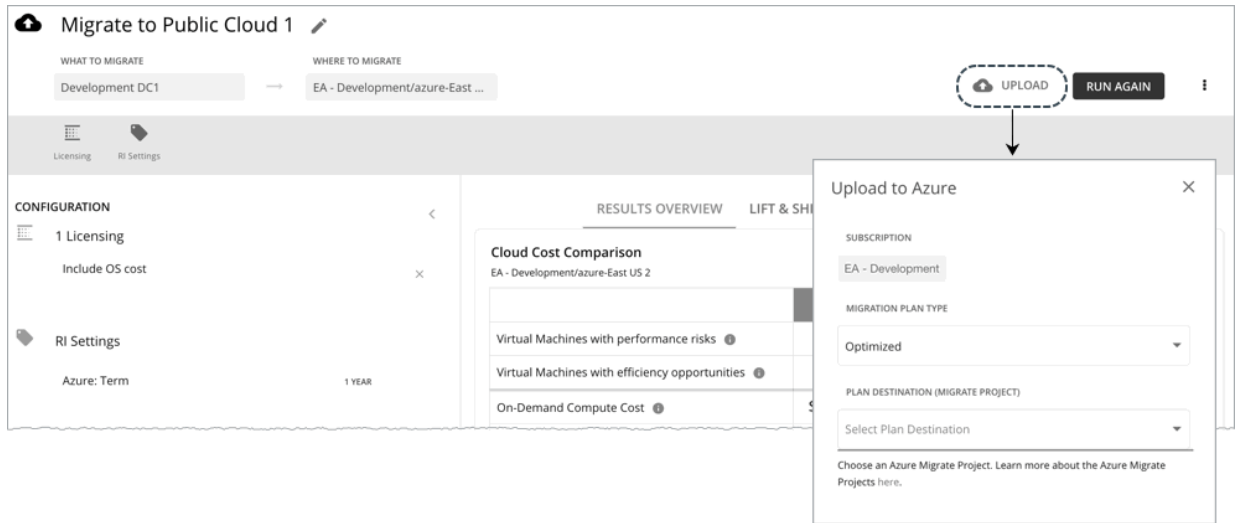
Before uploading the results, be sure to complete the following tasks:

1. Create a project in the Azure Migrate portal.
2. Add Turbonomic as a migration assessment solution to the project.
3. Set the necessary permissions in the Azure Migrate portal. For details about permissions, see [Azure Service Principal and Subscription Permissions](#).

Consult the Azure documentation for information on completing these tasks.

When you are ready to upload:

1. Click **Upload** at the top-right corner of the Plan Page.



2. Specify the following:

- Migration Plan Type

Choose to migrate either the 'Lift & Shift' or 'Optimized' results.

- Plan Destination (Migrate Project)

Select from the list of Azure Migrate projects. These are the projects belonging to the Azure subscription that you selected for the plan. If you have not created a project for the subscription, go to the Azure Migrate portal and create one.



**Warning:**

Uploading to a project with existing plan results overwrites those results.

The upload will fail if another upload targeted at the same destination is already in progress.

3. Click **Submit**.

The Plan Page updates to display the upload status. Refresh the page periodically to check:

- If the upload task completed without problems
- Any upload issues for individual entities

4. When the upload is complete, log in to the Azure Migrate portal and go to the project you selected as the plan destination.

The project should now display the uploaded information. Use the migration tools identified for the project to start the actual migration.

**Note:**


Repeat the upload procedure if you re-ran the plan and want to upload the new results.

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

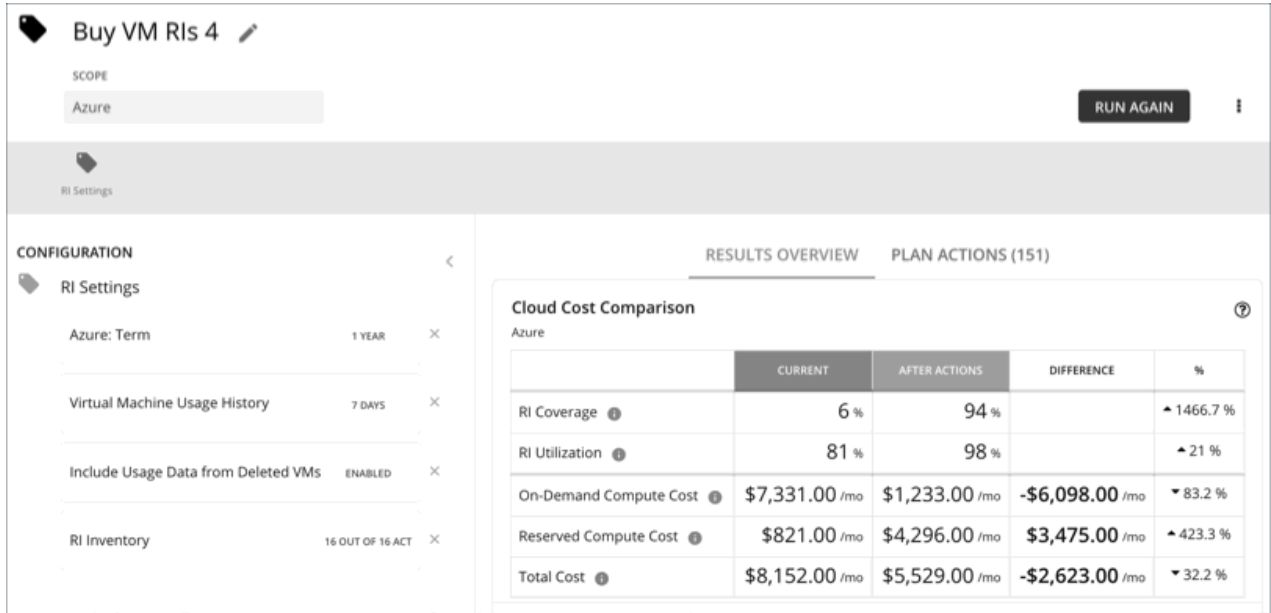
Use the toolbar on top of the Configuration section to change the configuration settings.

**Note:**

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Buy VM Reservations Plan



The screenshot shows the 'Buy VM RIs 4' interface. The 'SCOPE' is set to 'Azure'. A 'RUN AGAIN' button is visible in the top right. The 'CONFIGURATION' section on the left includes:
 

- RI Settings
- Azure: Term: 1 YEAR
- Virtual Machine Usage History: 7 DAYS
- Include Usage Data from Deleted VMs: ENABLED
- RI Inventory: 16 OUT OF 16 ACT

 The 'RESULTS OVERVIEW' section shows a 'Cloud Cost Comparison' table for Azure:
 

|                        | CURRENT        | AFTER ACTIONS  | DIFFERENCE      | %          |
|------------------------|----------------|----------------|-----------------|------------|
| RI Coverage            | 6 %            | 94 %           |                 | ▲ 1466.7 % |
| RI Utilization         | 81 %           | 98 %           |                 | ▲ 21 %     |
| On-Demand Compute Cost | \$7,331.00 /mo | \$1,233.00 /mo | -\$6,098.00 /mo | ▼ 83.2 %   |
| Reserved Compute Cost  | \$821.00 /mo   | \$4,296.00 /mo | \$3,475.00 /mo  | ▲ 423.3 %  |
| Total Cost             | \$8,152.00 /mo | \$5,529.00 /mo | -\$2,623.00 /mo | ▼ 32.2 %   |

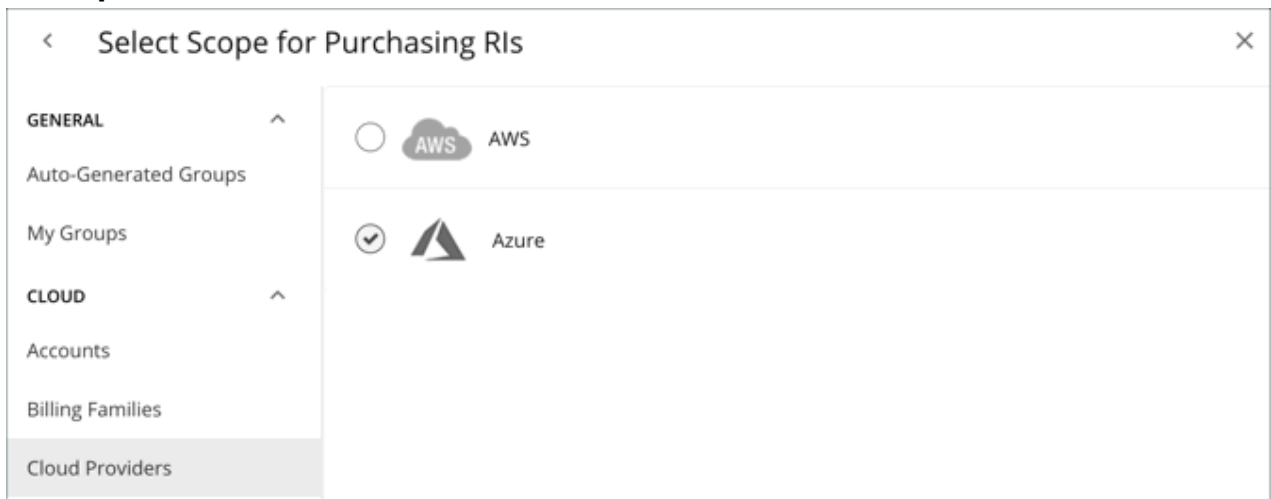
Run the Buy VM Reservations plan to see discount purchase opportunities that can significantly reduce on-demand costs for your cloud VMs. When calculating purchases, Turbonomic evaluates all purchasing options for your selected scope and usage data for the VMs in that scope. It then compares your current costs to the costs you would get after executing the plan recommendations.

Currently, Turbonomic can recommend purchase actions for AWS and Azure. Purchase actions for GCP will be introduced in a future release.

### Configuring a Buy VM Reservations Plan

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

#### 1. Scope



The screenshot shows the 'Select Scope for Purchasing RIs' dialog box. The 'GENERAL' section is expanded, showing 'Auto-Generated Groups' and 'My Groups'. The 'CLOUD' section is also expanded, showing 'Accounts', 'Billing Families', and 'Cloud Providers'. Under 'Cloud Providers', 'AWS' and 'Azure' are listed. 'AWS' is unselected, and 'Azure' is selected with a checkmark.

You can scope by:



- Accounts  
Choose AWS accounts or Azure subscriptions for the plan's scope.
- Billing Families  
Include discount purchases for a billing family. The plan calculates discount purchases through the billing family's master account.
- Cloud Providers  
See purchase opportunities for your AWS or Azure environment.
- Regions  
Focus the plan on a cloud provider's region.

## 2. RI Settings

<

### RI Settings

**Purchase RIs** ?

**AWS PROFILE**

OFFERING CLASS

Standard     Convertible

TERM

1 Year     3 Years

PAYMENT

All Upfront     Partial Upfront     No Upfront

**VIRTUAL MACHINE USAGE**

BASED ON THE PAST

1 DAY
7 DAYS
14 DAYS
30 DAYS

TERMINATED VIRTUAL MACHINES

Only use data from active VMs

Use data from active and deleted VMs (Support CI/CD pipeline)

**Discount Inventory** ?

This plan uses 18 out of 18 available RIs ✎ EDIT

### Purchase RIs

Allow the plan to buy discounts based on the following configurations:

- **Profile**

The settings that you have set up for real-time analysis are selected by default. You can change the settings to see how they affect costs.

- Offering Class

For AWS environments, choose the offering class that corresponds to the RI types that you typically use in your environment.

- Term

For AWS and Azure environments, choose the payment terms you contract for your discounts. TERM can be one of **1 Year** or **3 Year**. Typically, longer term payment plans cost less per year.

- Payment

The payment option that you prefer for your AWS RIs:

- All Upfront – You make full payment at the start of the RI term.
- Partial Upfront – You make a portion of the payment at the start of the term, with the remain cost paid at an hourly rate.
- No Upfront – You pay for the RIs at an hourly rate, for the duration of the term.

- **Virtual Machine Usage**

Specify the time frame you want the plan to use when it calculates your discount purchases.

- **Terminated Virtual Machines**

- **Only use data from active VMs** – Select this option if you terminate your VMs permanently.
- **Use data from active and deleted VMs (Support CI/CD pipeline)** – Select this option if you want to use data from a CI/CD pipeline that regularly deploys and terminates VMs.

### Discount Inventory

Select your discount inventory for the plan. You can use the default selection or any of the available discounts for your scope.

### Working With Buy VM Reservations Plan Results

After the Buy VM Reservations runs, you can view the results to see discount and optimization opportunities for your cloud environment.

The screenshot shows the 'Buy VM RIs 4' configuration page. On the left, under 'CONFIGURATION', the 'RI Settings' are visible: 'Azure: Term' is set to '1 YEAR', 'Virtual Machine Usage History' is '7 DAYS', 'Include Usage Data from Deleted VMs' is 'ENABLED', and 'RI Inventory' is '16 OUT OF 16 ACT'. The main area shows 'RESULTS OVERVIEW' with a 'Cloud Cost Comparison' table for Azure.

|                        | CURRENT        | AFTER ACTIONS  | DIFFERENCE      | %          |
|------------------------|----------------|----------------|-----------------|------------|
| RI Coverage            | 6 %            | 94 %           |                 | ▲ 1466.7 % |
| RI Utilization         | 81 %           | 98 %           |                 | ▲ 21 %     |
| On-Demand Compute Cost | \$7,331.00 /mo | \$1,233.00 /mo | -\$6,098.00 /mo | ▼ 83.2 %   |
| Reserved Compute Cost  | \$821.00 /mo   | \$4,296.00 /mo | \$3,475.00 /mo  | ▲ 423.3 %  |
| Total Cost             | \$8,152.00 /mo | \$5,529.00 /mo | -\$2,623.00 /mo | ▼ 32.2 %   |

### Viewing the Results

The plan results include the following charts:

- **Cloud Cost Comparison**

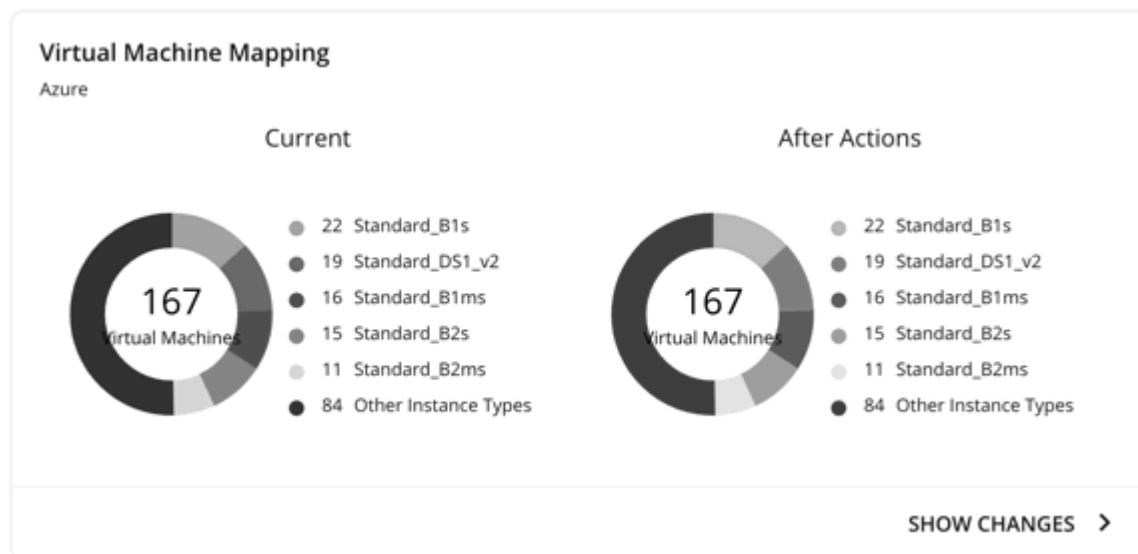
This chart highlights changes to your existing discount coverage and utilization if you execute all the actions that the plan recommends. Actions include increasing coverage or purchasing additional instance types at a discounted rate. Your cloud provider will adjust discount allocations when the actions have completed.

- Analysis evaluates ways to increase your current discount coverage so you can take full advantage of discounted pricing.
- The plan can recommend purchase actions to reduce your costs further. The analysis looks at historical VM usage and uptime to arrive at the number of instance types you should purchase.

You can compare current and after-action costs, including on-demand compute, discounted compute, and total costs. Purchase actions increase your discounted compute cost, but can lower your on-demand compute cost significantly as discount coverage increases. The end result is a reduction to your total cost.

### • **Virtual Machine Mapping**

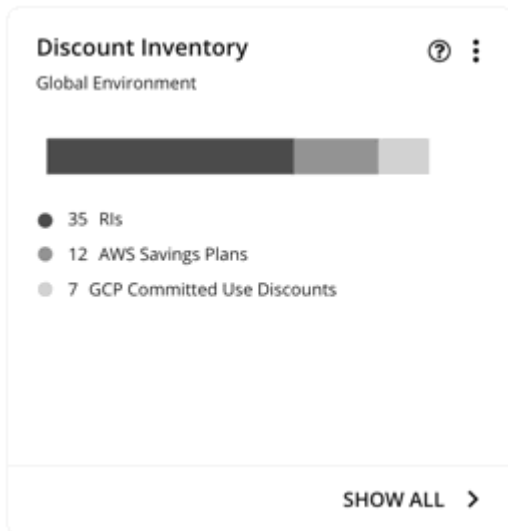
This chart shows the instance types for the VMs included in the plan.



Click **Show Changes** to see details for each VM with discount coverage changes. The table maps VMs to instance types, and shows how changes in discount coverage can reduce on-demand cost.

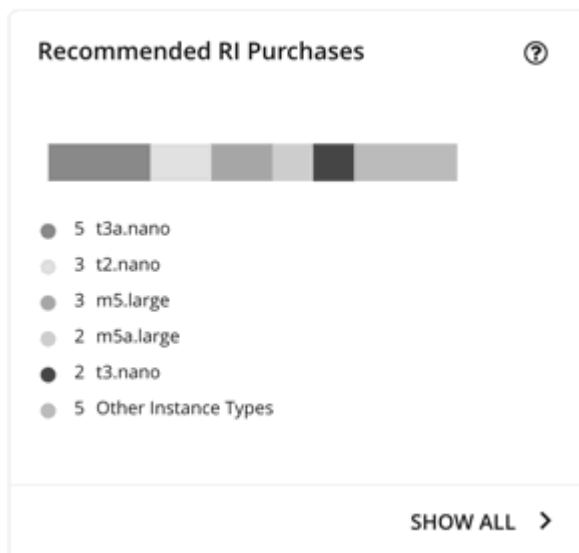
### • **Discount Inventory**

This chart lists the cloud provider discounts discovered in your environment. For a tabular listing, click **Show All** at the bottom of the chart. In the tabular listing, you can see if a discount expired before the specified purchase date.



- **Recommended RI Purchases**

Turbonomic can recommend purchasing instance types at a discounted rate to help you increase the percentage of VMs covered by discounted pricing and reduce on-demand costs. This chart shows your pending purchases. Download the list of purchases and then send it your cloud provider or representative to initiate the purchase process.



Click **Show All** to see a table with details for each discount.

The table shows the properties, up-front cost, and break-even period for each discount. The break-even period is the time at which savings will exceed the up-front cost, rounded to the month. The Cost Impact column indicates the monthly savings you would realize when you buy a specific discount.

When you choose one or more check boxes, the total count, up-front cost, and savings appear at the top.

## Viewing Plan Actions

Click the **Plan Actions** tab on top of the page to view a list of actions that you need to execute to achieve the plan results.

|                        |   | RESULTS OVERVIEW     |               | PLAN ACTIONS (274) |                   |                   |                       |                    |                 |               |         |
|------------------------|---|----------------------|---------------|--------------------|-------------------|-------------------|-----------------------|--------------------|-----------------|---------------|---------|
| DELETE                 | ^ | Scale Actions (28)   |               | TOTAL SAVINGS      | TOTAL INVESTMENTS |                   |                       |                    |                 |               |         |
|                        |   |                      |               | \$3,111.76/mo      | \$242.80/mo       |                   |                       |                    |                 |               |         |
| Volumes (131)          |   | Type to search       |               |                    |                   |                   |                       |                    | ADD FILTER      |               |         |
| SCALE                  | ^ | Virtual Machine Name | Instance Type | Discount Coverage  | On-Demand Cost    | New Instance Type | New Discount Coverage | New On-Demand Cost | Action Category | Cost Impact   | Action  |
| Volumes (79)           |   | eks-cluster-eks-cl   | m5a.4xlarge   | 0%                 | \$0.684/h         | r5a.2xlarge       | 75%                   | \$0.112/h          | SAVINGS         | ↓ \$417.49/mo | DETAILS |
| Virtual Machines (28)  |   | ocp47demo-2v5jc      | m5a.4xlarge   | 0%                 | \$0.684/h         | r5a.2xlarge       | 75%                   | \$0.112/h          | SAVINGS         | ↓ \$417.49/mo | DETAILS |
| Database Servers (7)   |   | cn-latest-lzdjn-woi  | m6i.2xlarge   | 0%                 | \$0.382/h         | r5a.xlarge        | 100%                  | \$0.00/h           | SAVINGS         | ↓ \$278.81/mo | DETAILS |
| BUY                    | ^ | cn-latest-lzdjn-woi  | m6i.2xlarge   | 0%                 | \$0.382/h         | r5.xlarge         | 50%                   | \$0.125/h          | SAVINGS         | ↓ \$187.33/mo | DETAILS |
| Reserved Instance (15) |   | pl-ocp-apm-gr2qp     | r5a.xlarge    | 0%                 | \$0.247/h         | r5.xlarge         | 100%                  | \$0.00/h           | PERFORMANCE     | ↓ \$180.07/mo | DETAILS |

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.


- RI Settings

Update your purchase settings to see how they impact results. For example, you can configure a longer timeframe so that the plan can include additional VM usage data in its analysis. For details, see [Purchase RIs](#).

- Discount Inventory

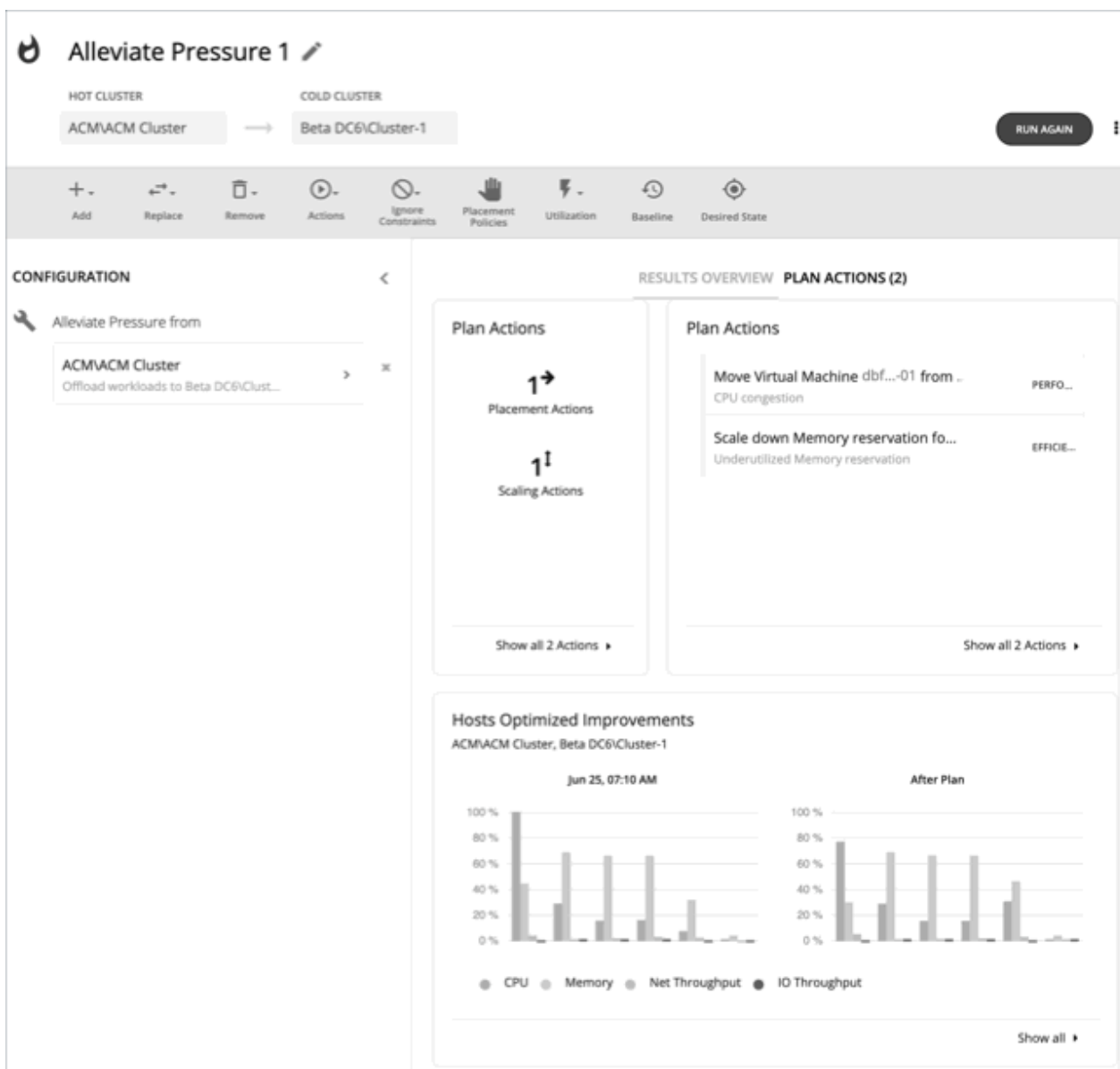
Use the default selection or any of the available discounts for your scope.

**Note:**

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Alleviate Pressure Plan



Use the Alleviate Pressure plan to find out how to migrate workloads from a stressed or *hot* cluster over to a cluster with more headroom. This plan shows the minimal changes you need to make to reduce risks on the hot cluster.

The plan results:

- Show the actions to migrate workloads from the hot cluster to the cold one
- Compare the current state of your clusters to the optimized state
- Show resulting headroom for both the hot and the cold clusters
- Show trends of workload-to-inventory over time for both clusters

Alleviate Pressure plans make use of the headroom in your clusters. Headroom is the number of VMs the cluster can support, for CPU, Memory and Storage.

To calculate cluster capacity and headroom, Turbonomic runs nightly plans that take into account the conditions in your current environment. The plans use the Economic Scheduling Engine to identify the optimal workload distribution for your clusters. This can include moving your current VMs to other hosts within the given cluster, if such moves would result in a more desirable workload distribution. The result of the plan is a calculation of how many more VMs the cluster can support.

To calculate VM headroom, the plan simulates adding VMs to your cluster. The plan assumes a certain capacity for these VMs, based on a specific VM template. For this reason, the count of VMs given for the headroom is an approximation based on that VM template.

To specify the templates these plans use, you can configure the nightly plans for each cluster. For more information, see [Configuring Nightly Plans](#)

**Note:**

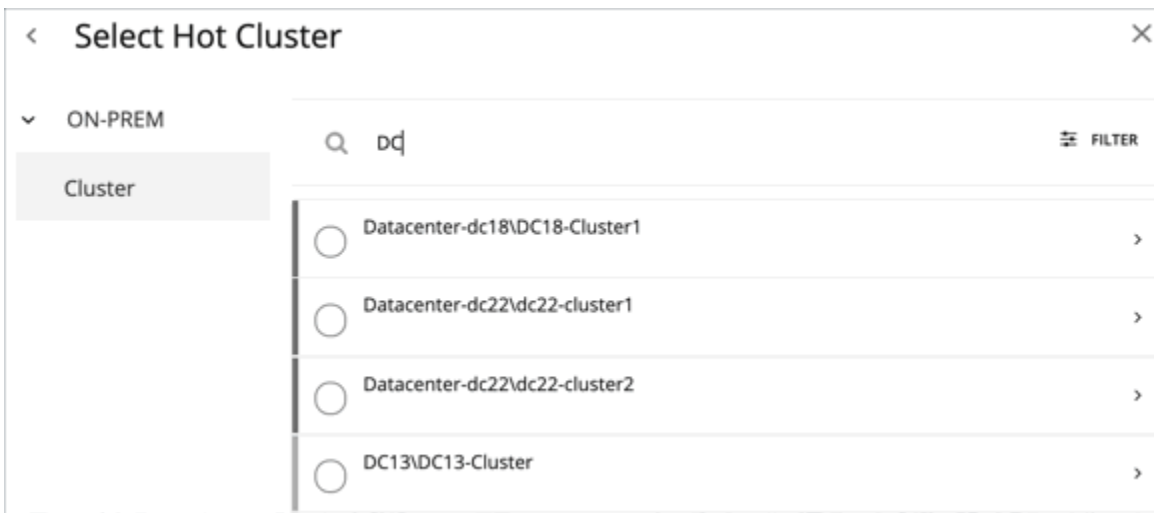
To execute, this plan must ignore certain constraints. The plan ignores cluster constraints to allow migrating workloads from the hot cluster to the cold one. It also ignores network constraints, imported DRS policies, and any Turbonomic that would ordinarily be in effect.

### ***Configuring an Alleviate Pressure Plan***

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

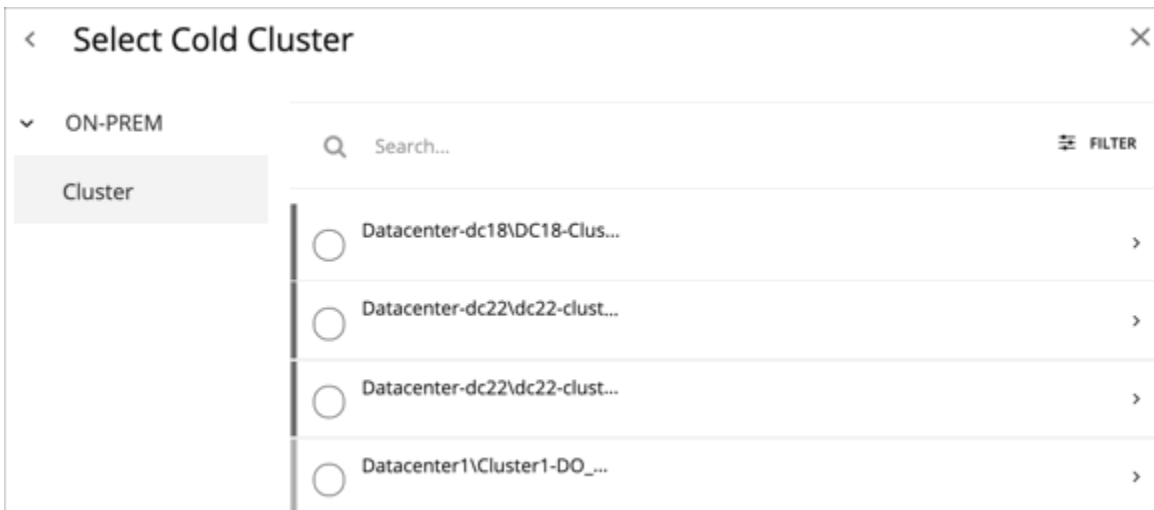
## **1. Scope**

The wizard first gives you a list for you to choose the hot cluster. This is the cluster that shows risks to performance. The list sorts with the most critical clusters first, and it includes the calculated headroom for CPU, Memory, and Storage in each cluster.



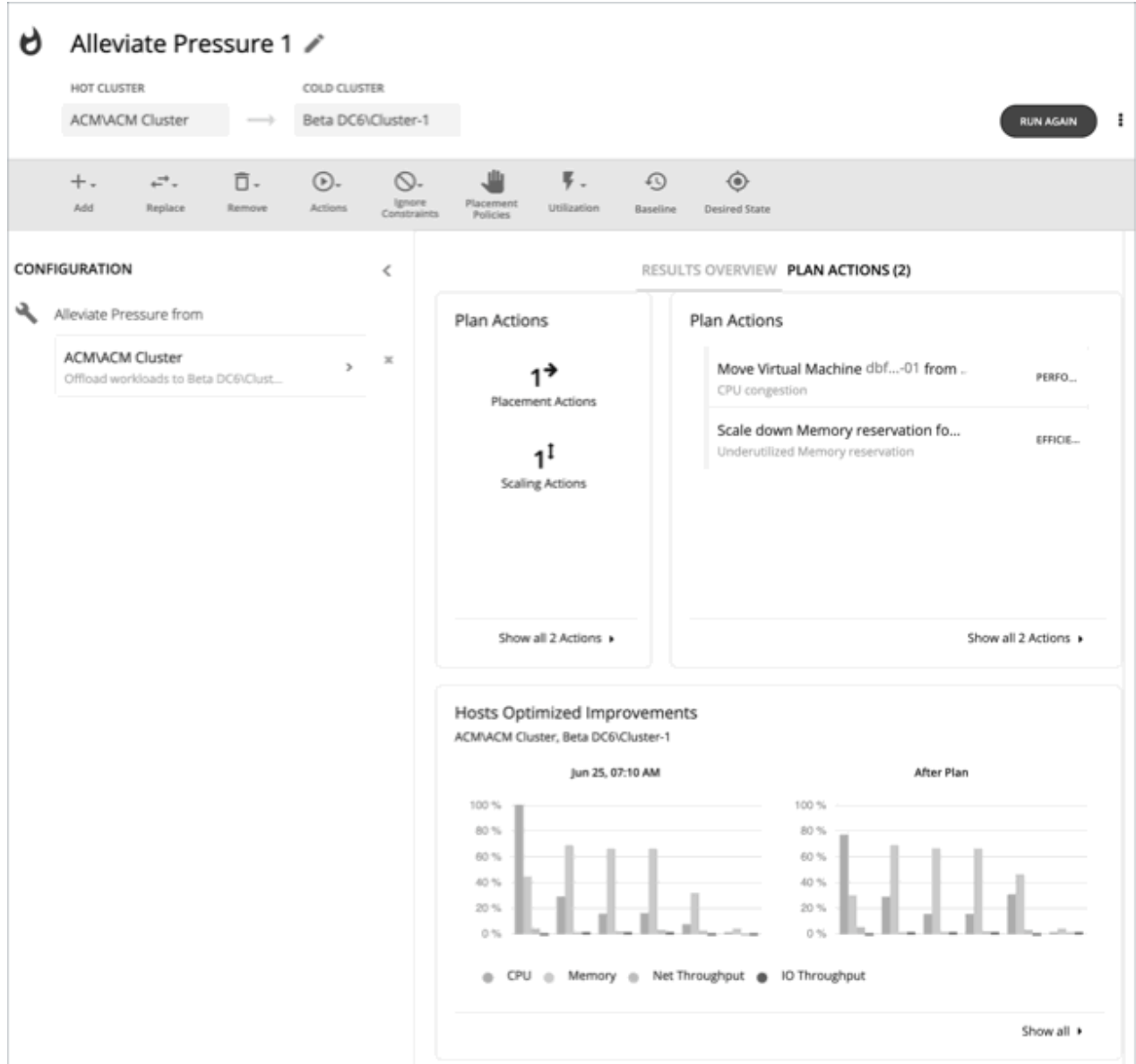
## **2. Cold Cluster**

After you select the hot cluster, choose the cold cluster.



## Working With Alleviate Pressure Plan Results

After the plan runs, you can view the results to see how the migration of workloads off of your hot cluster affects your environment.



## Viewing the Results

The results include the following charts:

- Plan Actions

You can see a list of actions to reduce the pressure on the hot cluster. It's typical to see actions to move workloads from the hot cluster over to the cold cluster. If some VMs are overprovisioned, you might see actions to reduce the capacity for those workloads.

- Hosts Optimized Improvements

This chart compares the current state of the hot cluster to its state after executing the plan actions. It displays the resource utilization of the cluster's hosts both before and after the plan.

- Headroom

With these charts, you can compare the headroom between the hot and cold clusters.



- Virtual Machines vs Hosts and Storage

This chart shows the total number of virtual machines, hosts, and storage in your on-prem environment, and tracks the data over time. Chart information helps you understand and make decisions around capacity and utilization, based on historical and projected demand.

## Re-Running the Plan

You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.



The toolbar items that display are similar to the toolbar items for a custom plan. For details, see [Configuring a Custom Plan](#).

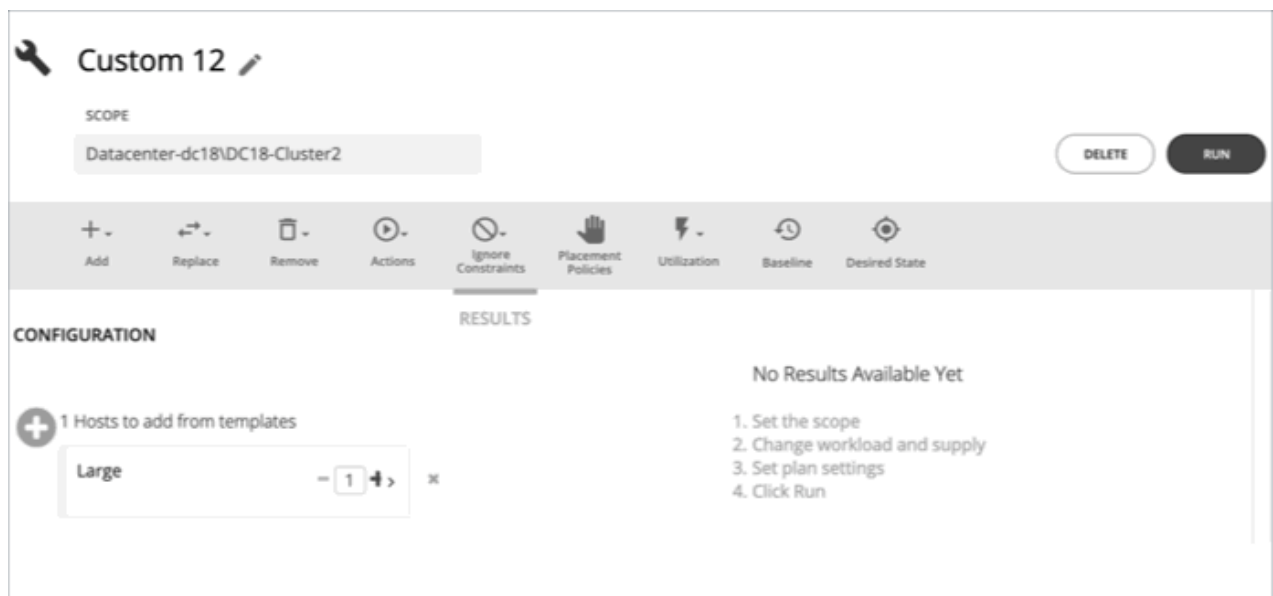
### Note:

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (⋮), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Custom Plan

For an overview of setting up plan scenarios, see [Settings Up User Plan Scenarios](#).



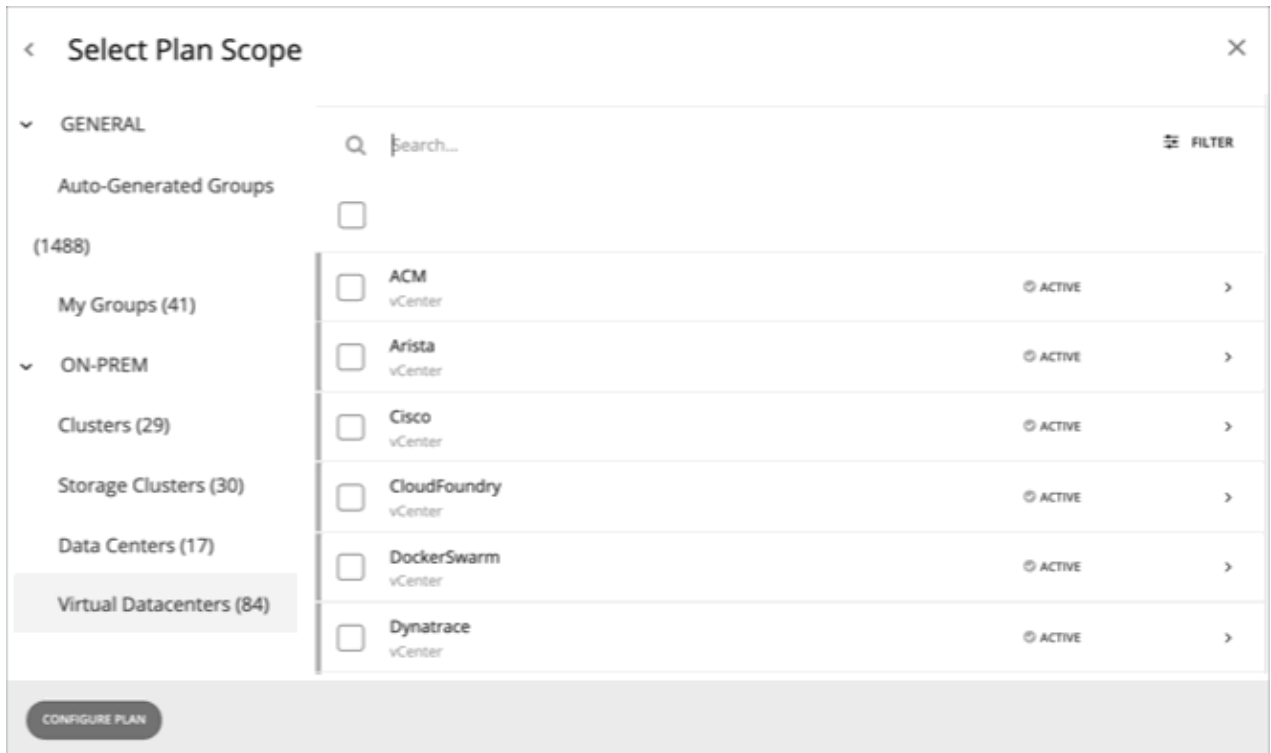
When you create a custom scenario, you specify the plan scope as an initial step, and then skip the plan wizards and jump straight into setting up the plan parameters. You can name the plan, change workload demand and the supply of resources, and specify other changes to the plan market.

## Configuring a Custom Plan

For an overview of setting up plan scenarios, see [Setting Up Plan Scenarios](#).

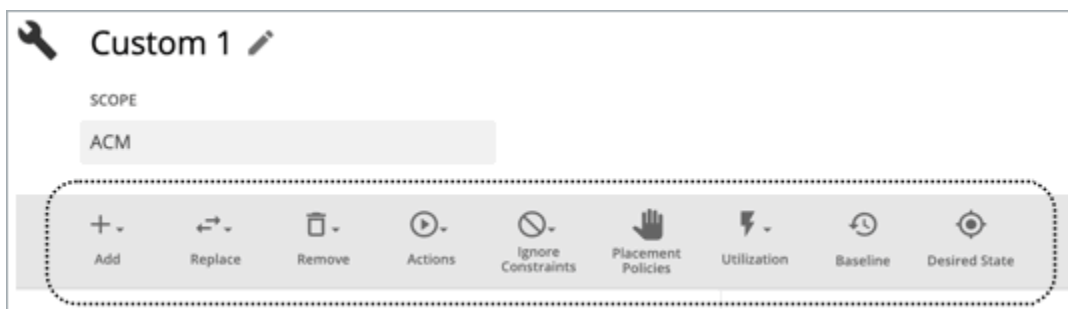
### 1. Scope

Specify the plan scope and then click **Configure Plan** at the bottom of the page.



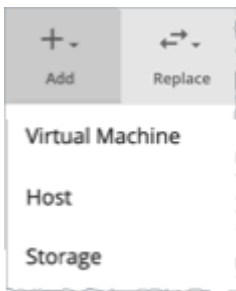
### 2. Plan Configuration

Use the Plan Configuration toolbar to fine-tune your plan settings. You can change workload demand and the supply of resources, and specify other changes to the plan market.



#### 2.1. Add

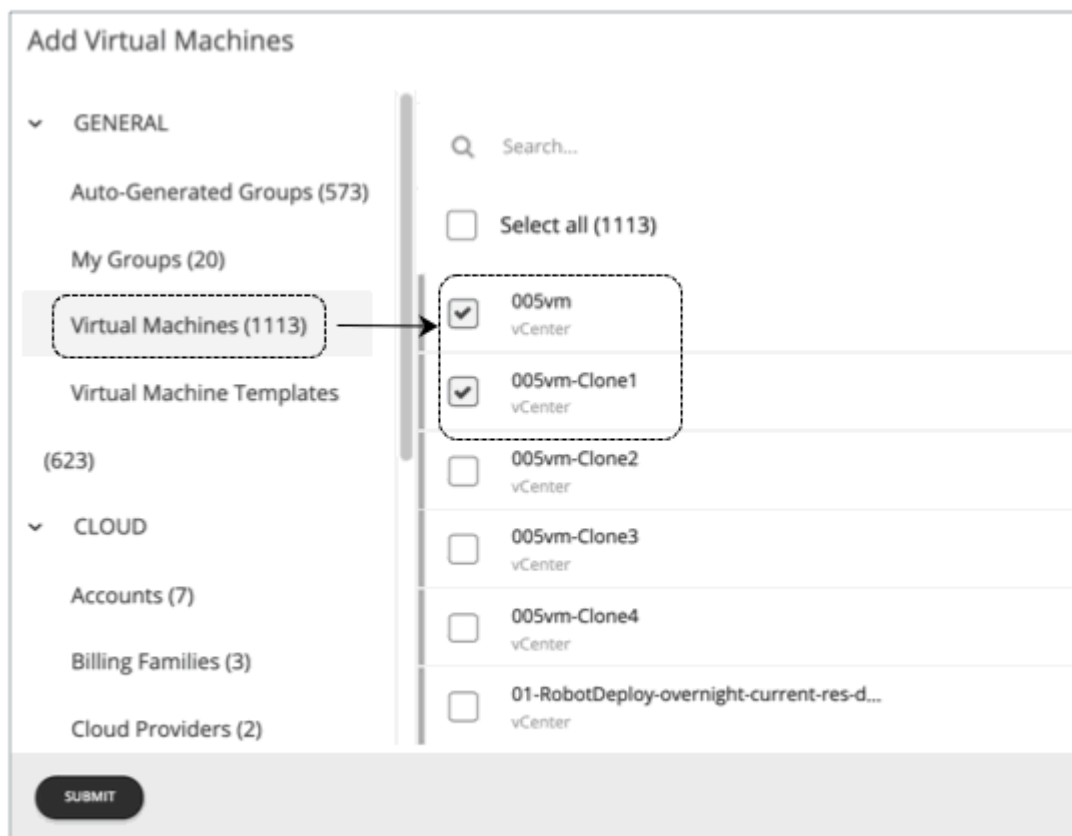
Add virtual machines, hosts, or storage to your plan. For example, when you add hosts, you increase the compute resources for the plan.



### Copy from an Entity or Template

Choose an entity or template to copy. This describes the new entities that Turbonomic will add to the plan. For example, you can run a plan that adds new VMs to a cluster. If you copy from a template, then the plan adds a new VM that matches the resource allocation you have specified for the given template.

- Option 1: Copy from an entity



- Option 2: Copy from a template

If no existing template is satisfactory, create one by clicking **New Template**.



**Note:**

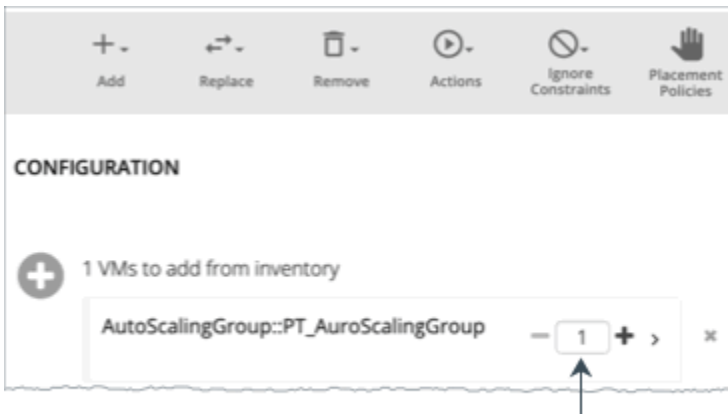
Turbonomic automatically adds any new template you create to the Template Catalog page (**Settings > Templates**).

It is not possible to use templates for containers or container pods.

Use the **Filter** option to show entities or templates with certain properties (name, number of CPUs, etc.). This makes it easier to sort through a long list.

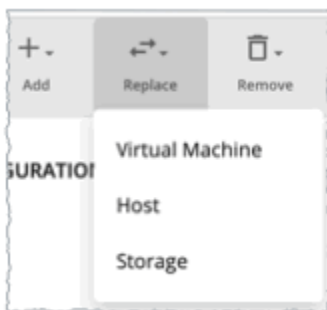
**Number of Copies to Add**

After choosing an entity or template, it appears as an entry in the Configuration summary. Then you can set how many copies to add.



Set how many copies to add

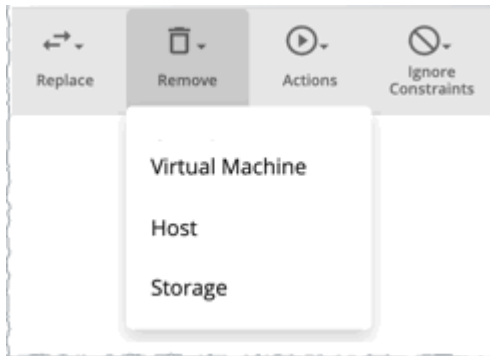
**2.2. Replace**



Replacing virtual machine is a way to change the properties of VMs in your plan market. When you replace workload, you select one or more VMs that you want to change, and then you select a template to use in their place. The list of changed VMs displays in the Configuration Summary. You can delete individual entries from the this summary if necessary.

Replacing hosts or storage is a way to plan for a hardware upgrade. For example, if you replace your hosts or datastores with a more powerful template, the plan might show that you can use fewer hosts or datastores, and it will show the best placement for workloads on those entities. You begin by selecting the entities you want to replace, and when you click **REPLACE** you can then choose a template that will replace them. Note that you can only choose a single template for each set of entities you want to have replaced. You can configure different replacements in the same plan, if you want to use more than one template.

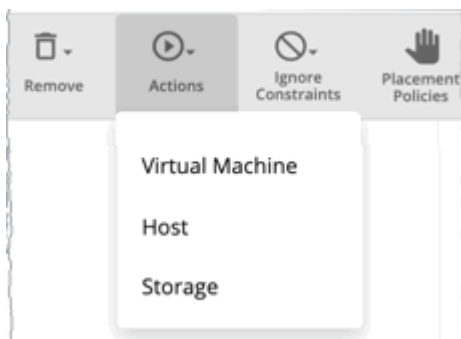
### 2.3. Remove



Removing virtual machines frees up resources for other workloads to use.

Removing hosts or storage means you have fewer compute or storage resources for your workloads. If you think you have overprovisioned your environment, you can run a plan to see whether fewer hosts or less storage can still support the same workload.

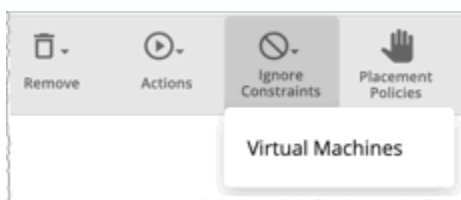
### 2.4. Actions



See the effect of enabling or disabling actions on the entities included in the plan. For example, you might plan for more workload but know that you don't want to add more hardware, so you disable Provision of hosts for your plan. The results would then indicate if the environment can support the additional workload.

### 2.5. Ignore Constraints

Choose to ignore constraints (such as placement policies) for VMs in your environment.



By default, VMs are constrained to the cluster, network group, datacenter, or storage group that their hosts belong to. You can choose to ignore these boundaries.

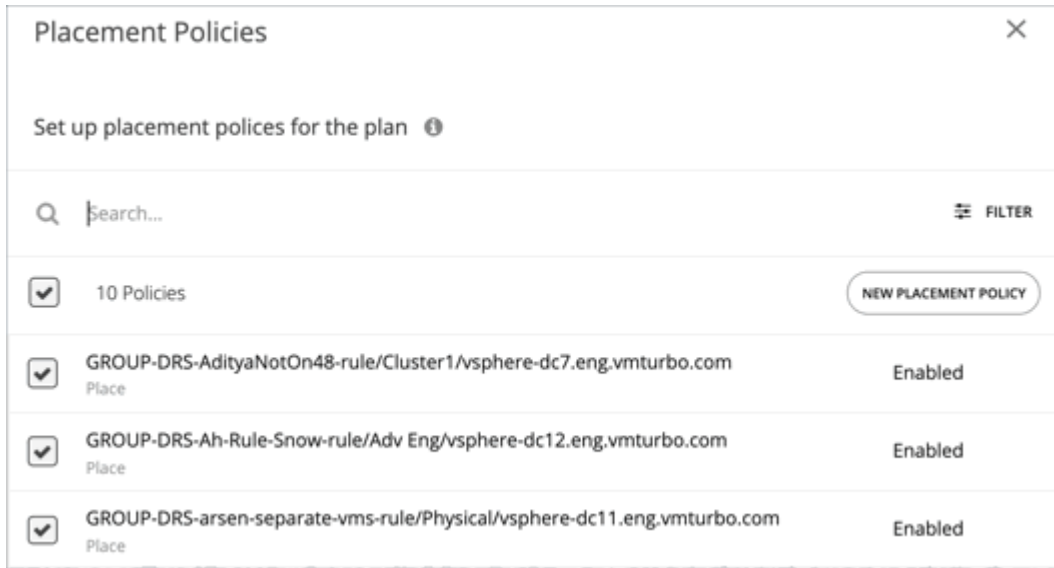
For example, by default a plan does not consider moving VMs to physical hosts outside of the current cluster. If you disable the Cluster constraint for a VM in your plan, then the plan can evaluate the results of hosting those VMs on any other physical machine within the scope of your plan. If the best results come from moving that VM to a different cluster, then the plan will show that result.

**Note:**

If you are adding hosts to a plan, and use host templates, then you must turn on **Ignore Constraints**.

## 2.6. Placement Policies

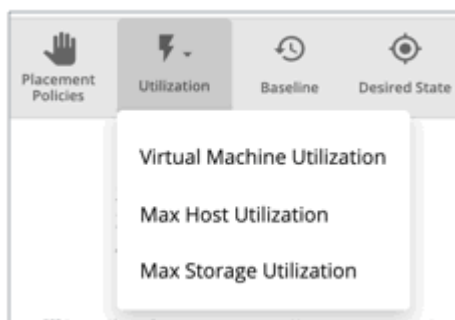
By default, the plan includes all the placement policies that apply to the plan scope. Also, these policies are in their real-time state (enabled or disabled).



You can use these settings to enable or disable existing policies, or you can create new policies to apply only to this plan scenario. For information about creating placement policies, see [Placement Policies](#).

## 2.7. Utilization

Setting utilization by a certain percentage is a way to increase or decrease the workload for the scope of your plan and any entity added to the plan, or for specific groups. Turbonomic uses the resulting utilization values as the baseline for the plan.



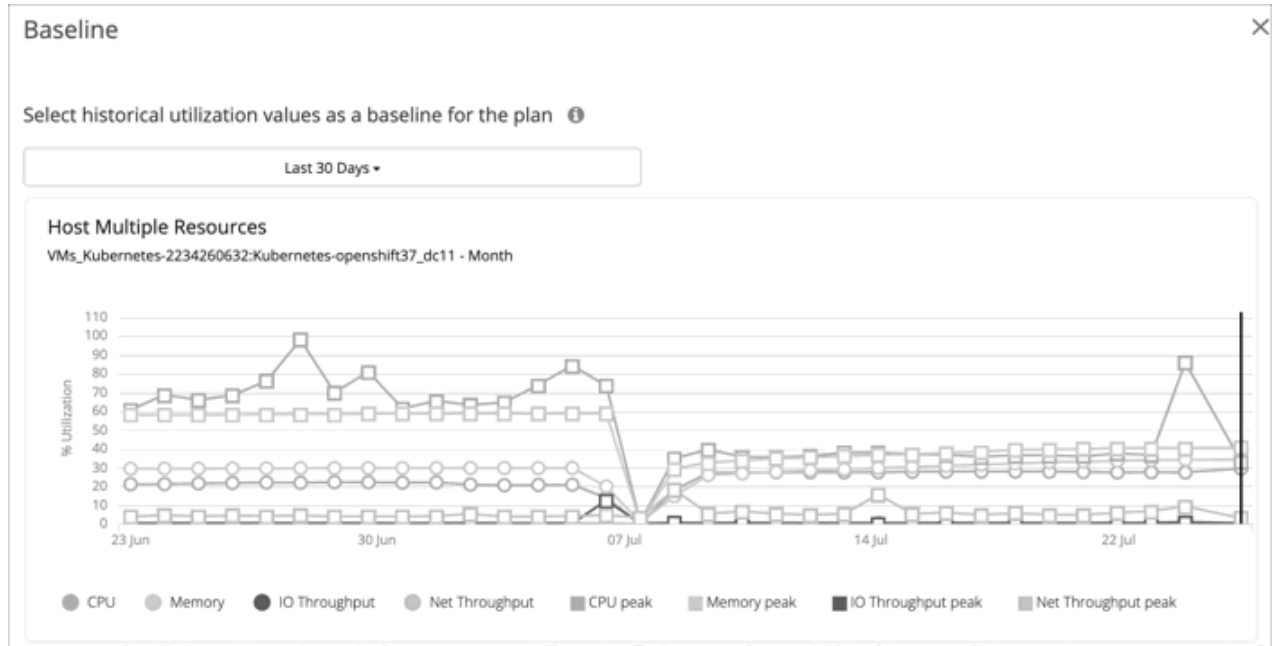
Max Host Utilization levels specify the percentage of the physical resource that you want to make available in the given plan. By default, hosts have utilization set to 100%. For a given plan, you can set the utilization to a lower value. For example, assume you want to simulate High Availability of 25% for some hosts in the plan. In that case, you can select these hosts and set their utilization levels to 75%.

Max Storage utilization levels specify the percentage of the physical resource that you want to make available in the given plan. By default, storage has utilization set to 100%. For a given plan, you can set the utilization to a lower value. For example, assume you have one data store that you want to share

evenly for two clusters of VMs. Also assume that you are creating a plan for one of those clusters. In that case, you can set the datastores to 50% utilization. This saves storage resources for the other cluster that will use this storage.

## 2.8. Baseline

Use these settings to set up the baseline of utilization metrics for your plan.

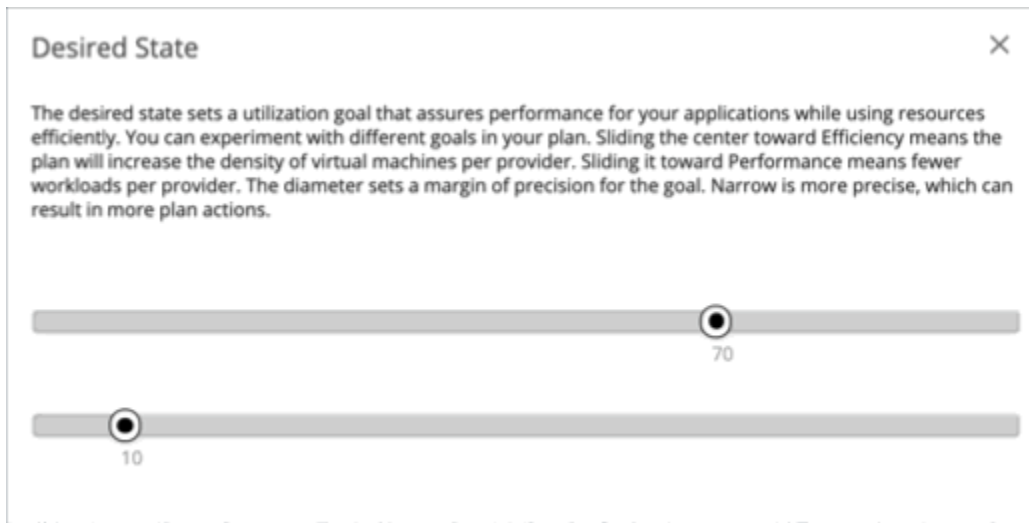


By default, the plan runs against the current state of your environment. You can set up the plan to add or remove entities, or otherwise affect the plan calculations. But the utilization metrics will be based on the current state of the plan. If you run the same plan multiple times, each run begins with a fresh view of your inventory.

You can select from the list of snapshots to load the utilization statistics from a previous time period into the plan. Use this to run the plan against utilization that you experienced in the past. For example, assume a peak utilization period for the month before the winter holidays. During the holidays you want to plan to add new capacity that can better handle that peak. You would set the baseline to the utilization you saw during that pre-holiday peak.

## 2.9. Desired State

The desired state is a condition in your environment that assures performance for your workloads, while it utilizes your resources as efficiently as possible and you do not overprovision your infrastructure. Turbonomic uses default Desired State settings to drive its analysis. You should never change the settings for real-time analysis unless you are working directly with Technical support. However, you can change the settings in a plan to see what effect a more or less aggressive configuration would have in your environment.



You can think of the desired state as an n-dimensional sphere that encompasses the fittest conditions your environment can achieve. The multiple dimensions of this sphere are defined by the resource metrics in your environment. Metric dimensions include VMem, storage, CPU, etc. While the metrics on the entities in your environment can be any value, the desired state, this n-dimensional sphere, is the subset of metric values that assures the best performance while achieving the most efficient utilization of resources that is possible.

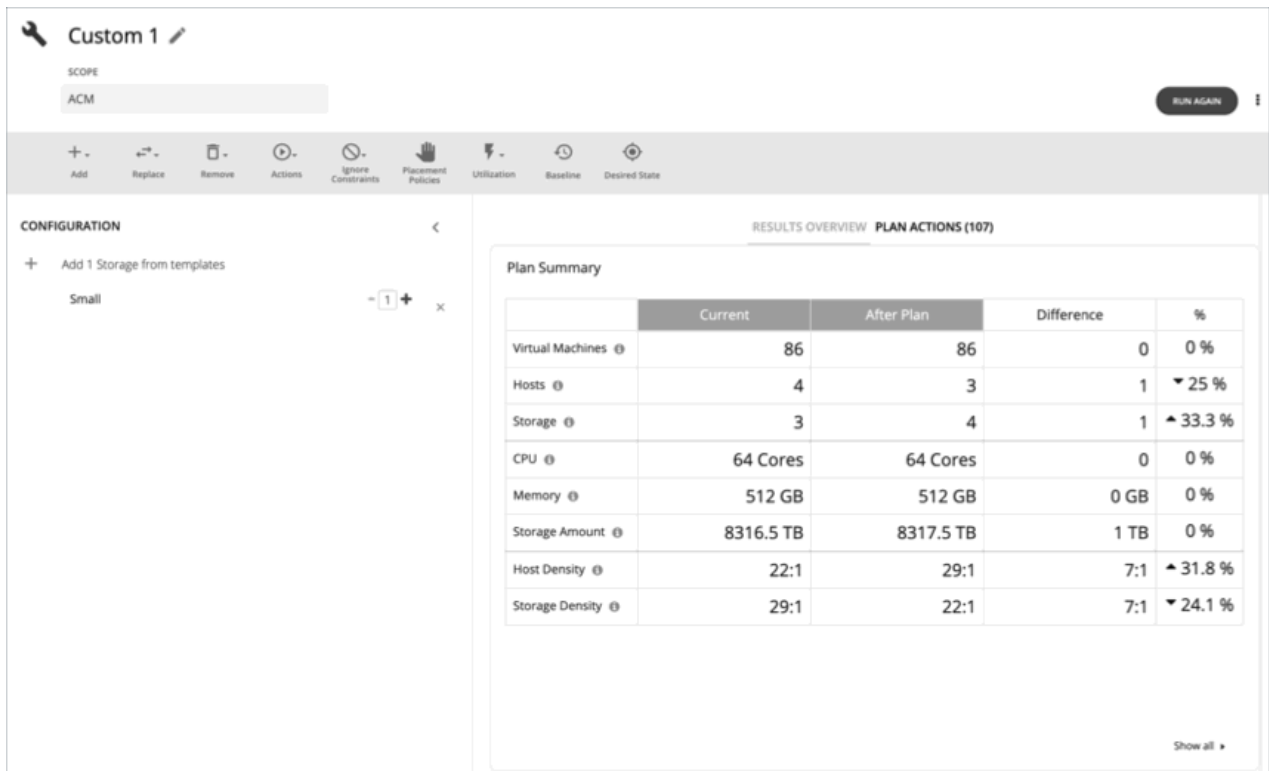
The Desired State settings center this sphere on Performance (more infrastructure to supply the workload demand), or on Efficiency (less investment in infrastructure to supply the workload demand). The settings also adjust the diameter of the sphere to determine the range of deviation from the center that can encompass the desired state. If you specify a large diameter, Turbonomic will have more variation in the way it distributes workload across hosting devices.

For more information, see [The Desired State](#).

### ***Working With Custom Plan Results***

After the plan runs, you can view the results to see how the plan settings you configured affect your environment.





## Viewing the Results

The results include the following charts:

- **Plan Summary Chart**

This chart compares your current resources to the resources you would get after executing the plan.

**Note:**

Under some circumstances, this chart might not count "non-participating" entities in the real-time market, such as suspended VMs or hosts in a failover state. The following charts, on the other hand, count all entities in the real-time market, regardless of state:

- Scope Preview chart (displays before you run the plan)
- Optimized Improvements and Comparison charts

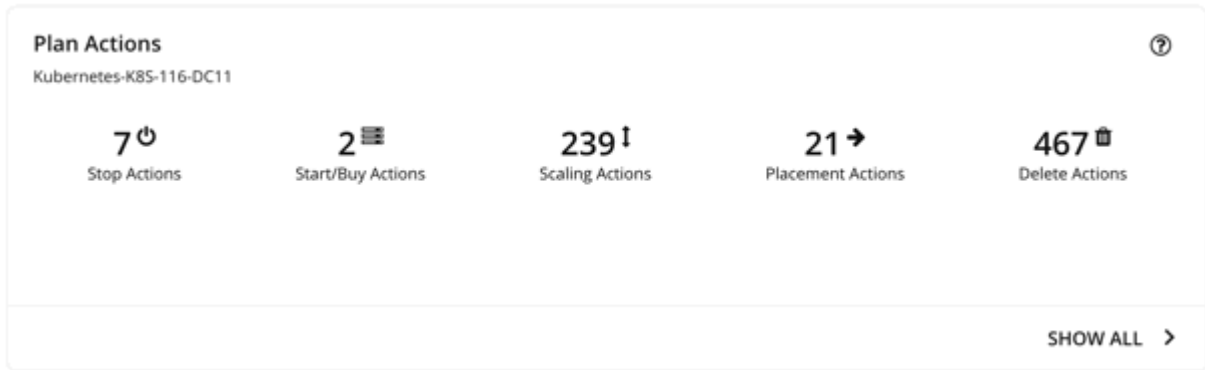
If the plan's scope includes VMs that cannot be placed, the results include a notification indicating the number of VMs. Click **Show Details** to see the list of VMs and the reasons for their non-placement.

Click **Show all** at the bottom of the chart to see savings or investment costs, or to download the chart as a CSV file.

- **Plan Actions Chart**

This chart summarizes the actions that you need to execute to achieve the plan results. For example, if you run an Alleviate Pressure plan, you can see actions to move workloads from the hot cluster over to the cold cluster. If some VMs are overprovisioned, you might see actions to reduce the capacity for those workloads.

The text chart groups actions by action type. The list chart shows a partial list of actions.

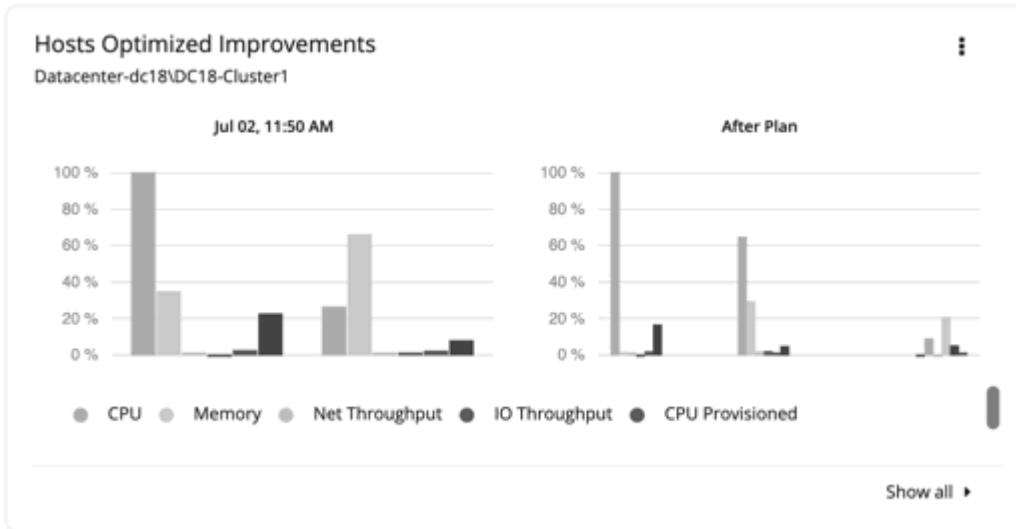


To view action details or download the list of actions as a CSV file:

- Click an action type in the text chart or an individual action in the list chart.
- Click **Show All** at the bottom of the chart.

• **Optimized Improvements Charts for Hosts, Storage, and Virtual Machines**

The Optimized Improvements chart shows how the utilization of resources would change assuming you accept all of the actions listed in the Plan Actions chart.



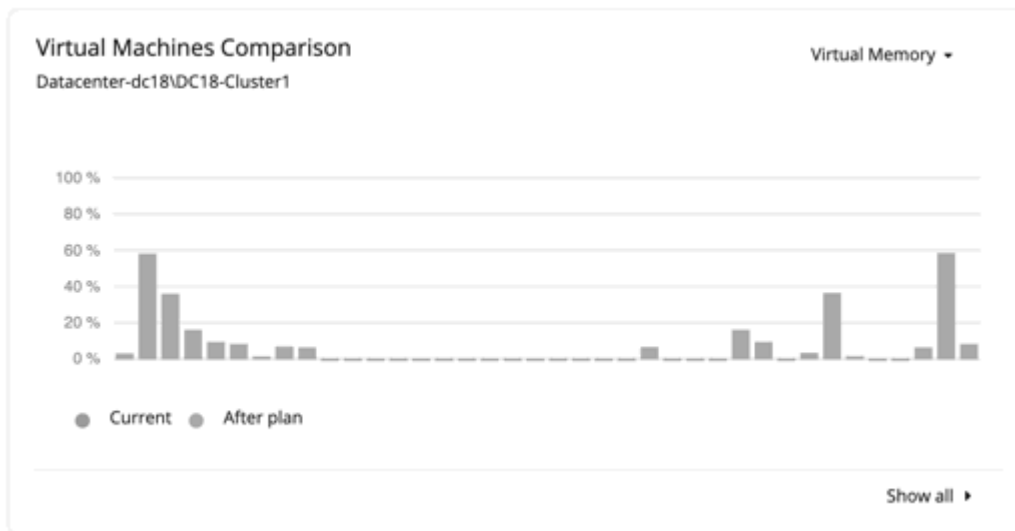
- In many of these charts, you can change the commodities on display. To do this, go to the top-right section of the chart, click the More options icon (⋮), and then select **Edit**. In the new screen that displays, go to the **Commodity** section and then add or remove commodities.

To restore the default commodities, use the **Reset view** option at the top-right section of the page.


- Click **Show all** at the bottom of the chart to see a breakdown of the current chart data by entity (for example, show CPU, Memory, and IO Throughput utilization for each host), or to download chart data as a CSV file.

• **Comparison Charts for Hosts, Storage Devices, and Virtual Machines**

A Comparison chart shows how the utilization of a particular commodity (such as memory or CPU) for each entity in the plan would change if you execute the actions listed in the Plan Actions chart.



- To change the commodity displayed in the chart, go to the top-right section of a chart and then select from the list of commodities.

To restore the default commodity, go to the top-right section of the page, click the More options icon (  ), and then select **Reset view**.

- Click **Show all** at the bottom of the chart to show a breakdown of the current chart data by entity (for example, show Virtual Memory utilization for each virtual machine), or to download the chart as a CSV file.

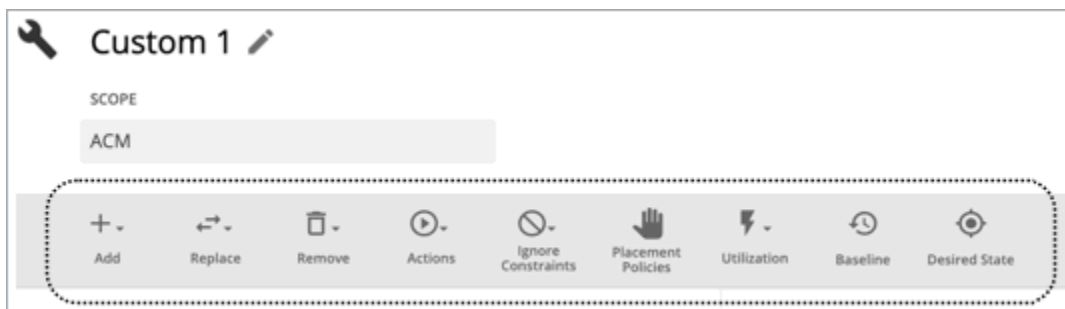
**Note:**

For the Storage Devices Comparison chart, if you set the view to **VM Per Storage** and click **Show all**, the total number of VMs sometimes does not match the number in the Plan Summary chart. This happens if there are VMs in the plan that use multiple storage devices. The Storage Devices Comparison chart counts those VMs multiple times, depending on the number of storage devices they use, while the Plan Summary chart shows the actual number of VMs.

## Re-Running the Plan


You can run the plan again with the same or a different set of configuration settings. This runs the plan scenario against the market in its current state, so the results you see might be different, even if you did not change the configuration settings.

Use the toolbar on top of the Configuration section to change the configuration settings.



For details about these settings, see [Configuring a Custom Plan](#).

**Note:**

It is not possible to change the scope of the plan in the Plan Page. You will need to start over if you want a different scope. To start over, go to the top-right section of the page, click the More options icon (  ), and then select **New Plan**.

When you are ready to re-run the plan, click **Run Again** on the top-right section of the page.

## Configuring Nightly Plans

The plan and its template

Expand to see template details

|                                                                                         |                         |                             |
|-----------------------------------------------------------------------------------------|-------------------------|-----------------------------|
| 0 MB/s                                                                                  | 0 MB/s                  | 1                           |
| IO THROUGHPUT SIZE                                                                      | NETWORK THROUGHPUT SIZE | NUM OF CPU                  |
| 3465 MHz                                                                                | 0 %                     | 1243.0769 MB                |
| CPU SPEED                                                                               | CPU CONSUMED FACTOR     | MEMORY SIZE                 |
| 0 %                                                                                     | 0 MB/s                  | 0 MB/s                      |
| MEMORY CONSUMED FACTOR                                                                  | IO THROUGHPUT CONSUMED  | NETWORK THROUGHPUT CONSUMED |
| 1.344238 GB                                                                             | 0                       | 20.701876 %                 |
| DISK SIZE                                                                               | DISK IOPS CONSUMED      | DISK CONSUMED FACTOR        |
| AVG:GROUP-PMsByCluster_vsphere-dc18.corp.vmturbo.com_datacenter-21_domain-c567_HEADROOM |                         |                             |
| MODEL                                                                                   |                         |                             |

Turbonomic runs nightly plans to calculate headroom for the clusters in your on-prem environment. For each cluster plan, you can set which VM template to use in these calculations.

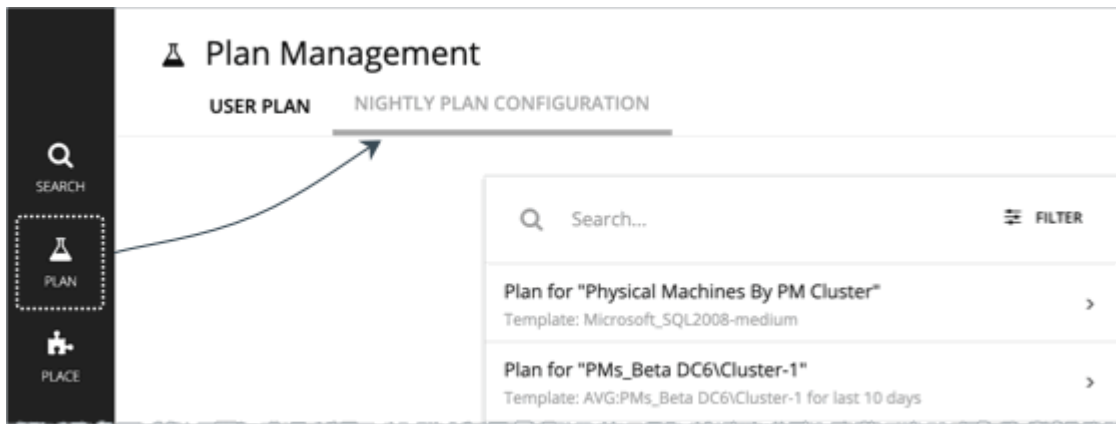
For information about viewing cluster headroom, see [Viewing Cluster Headroom](#).

To calculate cluster capacity and headroom, Turbonomic runs nightly plans that take into account the conditions in your current environment. The plans use the Economic Scheduling Engine to identify the optimal workload distribution for your clusters. This can include moving your current VMs to other hosts within the given cluster, if such moves would result in a more desirable workload distribution. The result of the plan is a calculation of how many more VMs the cluster can support.

To calculate VM headroom, the plan simulates adding VMs to your cluster. The plan assumes a certain capacity for these VMs, based on a specific VM template. For this reason, the count of VMs given for the headroom is an approximation based on that VM template.

To set templates to use for the nightly plans:

1. Navigate to the Plan Page and click **NIGHTLY PLAN CONFIGURATION**.



This displays a list of all the nightly plans. Turbonomic creates a nightly plan for each cluster.

2. Click the plan that you want to configure.

A fly-out appears that lists all the available templates.

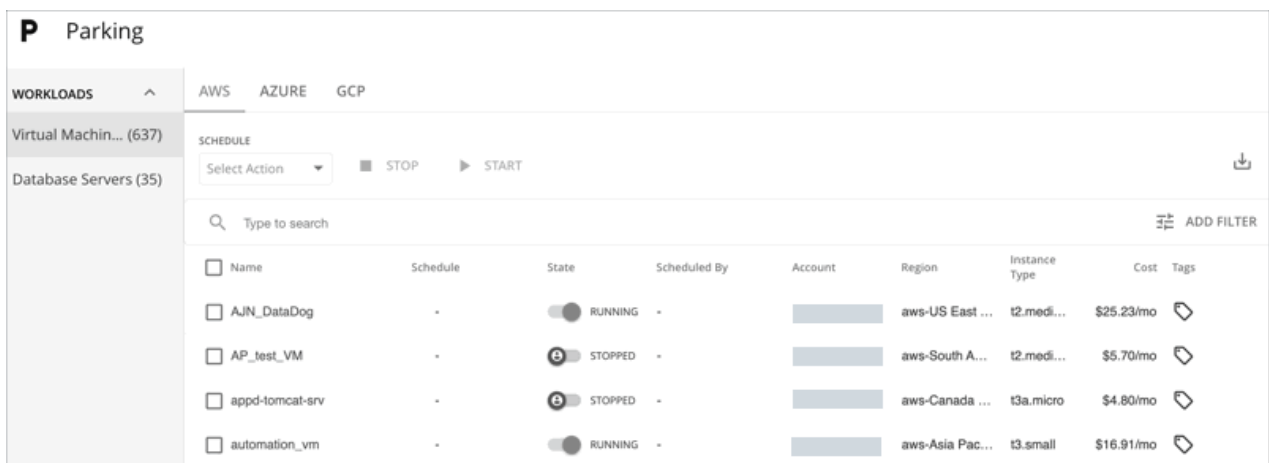
3. Select the template you want for this plan.

Choose the template and click **Select**.

## Park: Stop or Start Cloud Resources

Turbonomic supports 'parking' actions for cloud resources. These actions *stop* your cloud resources for a given period of time to help you reduce your cloud expenses, and then *start* these resources later when you need them. You can enforce parking actions on demand or according to a schedule.

When you click **Park** on the main menu, the user interface displays a list of parkable entities discovered from cloud targets.



The following table describes the current level of support for parking actions.

| Supported Entity | Parkable                                       | Not Parkable                                                         |
|------------------|------------------------------------------------|----------------------------------------------------------------------|
| VM               | VMs discovered via AWS, Azure, and GCP targets | VMs on consistent scaling groups, auto scaling groups, or scale sets |
| Database Server  | AWS RDS instances<br><b>Note:</b>              | AWS RDS clusters and cluster members                                 |

| Supported Entity | Parkable                                                                                                                                                                                                   | Not Parkable |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
|                  | AWS provides guidelines and caveats for parking actions. For example, your accounts will continue to incur certain charges while instances are stopped. For details, visit this <a href="#">AWS page</a> . |              |

## Permissions

Be sure that your cloud targets have the required permissions so that Turbonomic can enforce parking actions. For a full list of permissions, see the following topics:

- [AWS Permissions](#)
- [Azure Permissions](#)
- [GCP Permissions](#)

## How Parking Works

Cloud VMs and Database Servers that become idle for a period of time (for example, after business hours) and those that undergo regular maintenance are ideal for parking. When parking actions are enforced, Turbonomic reverses any conflicting actions on these entities. For example, when a VM has been started via Turbonomic and a user stops the VM via the cloud provider, Turbonomic will start the VM after a subsequent discovery cycle.

When a scaling action is in progress, Turbonomic postpones a stop or start action until the scaling action completes.

## Working with the Parking Page

After discovering parkable cloud VMs and Database Servers, Turbonomic adds them to the Parking page and shows their state (stopped or running) at the time of discovery. The page shows VMs and Database Servers as separate groups, so you can manage them efficiently. Use the page to execute parking actions or attach parking schedules to these entities.

**P Parking**

WORKLOADS ^ AWS AZURE GCP

Virtual Machines (637)

Database Servers (35)

SCHEDULE

Select Action [STOP] [START]

Type to search

| Name            | Schedule | State   | Scheduled By | Account | Region          | Instance Type | Cost       | Tags |
|-----------------|----------|---------|--------------|---------|-----------------|---------------|------------|------|
| AuN_DataDog     | -        | RUNNING | -            |         | aws-US East ... | t2.medi...    | \$25.23/mo |      |
| AP_test_VM      | -        | STOPPED | -            |         | aws-South A...  | t2.medi...    | \$5.70/mo  |      |
| appd-tomcat-srv | -        | STOPPED | -            |         | aws-Canada ...  | t3a.micro     | \$4.80/mo  |      |
| automation_vm   | -        | RUNNING | -            |         | aws-Asia Pac... | t3.small      | \$16.91/mo |      |

## Points to consider

- Users with the Site Administrator, Administrator, or Automator role can manage the parking actions and settings in the Parking page. Users with other roles have read-only access to the page. For details about roles, see [Managing User Accounts](#).

- Parking actions can only be executed from the Parking page. The Pending Actions, Potential Savings, and Top Accounts charts in the Home Page only show the non-parking actions that you can execute.
- Add the All Actions chart to your dashboard and use the tabular view to see and download a list of parking (and non-parking) actions that have been executed successfully.
- Currently, actual savings associated with parking actions are not reflected in the Parking page or the Cumulative Savings chart.

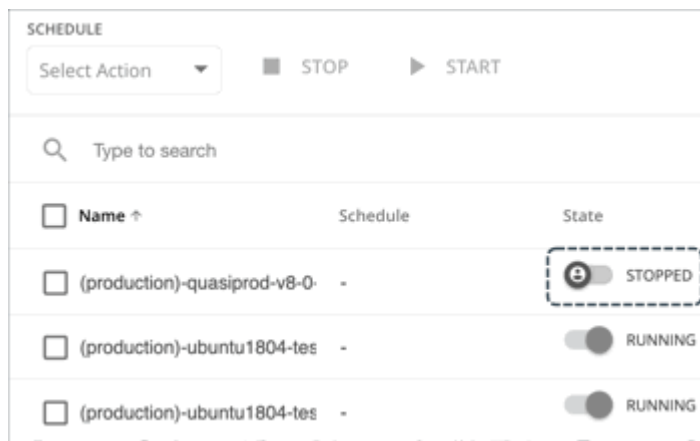
**Note:**

For VMs, the **Cost** column in the Parking page shows the estimated on-demand monthly cost for a VM, for your reference.

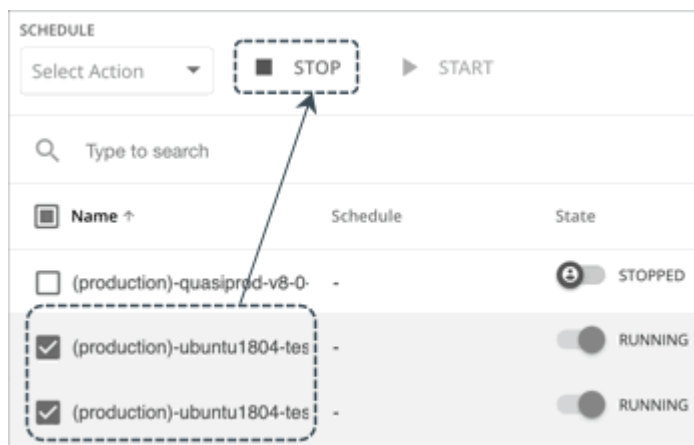
## Executing Parking Actions

You can execute parking actions in the following ways:

- For individual entities, use the toggle under the **State** column to change the currently enforced parking action.



- Select one or several entities with the same state, and then click **Stop** or **Start**.



In the All Actions chart, the tabular view shows the following information for the parking actions that you executed:

| All Actions             |                                              |                        |                                    |                         |
|-------------------------|----------------------------------------------|------------------------|------------------------------------|-------------------------|
| API-Test - Month        |                                              |                        |                                    |                         |
| DATE CREATED            | ACTION DESCRIPTION                           | RISK TYPE              | EXECUTION                          | DATE EXECUTED           |
| 01 Mar 2023<br>02:31 PM | Start Virtual Machine API-Test in [REDACTED] | Efficiency Improvement | Manually Accepted By administrator | 01 Mar 2023<br>02:31 PM |

- Execution: Manually Accepted
- Risk Type: Savings (stop actions) or Efficiency Improvement (start actions)

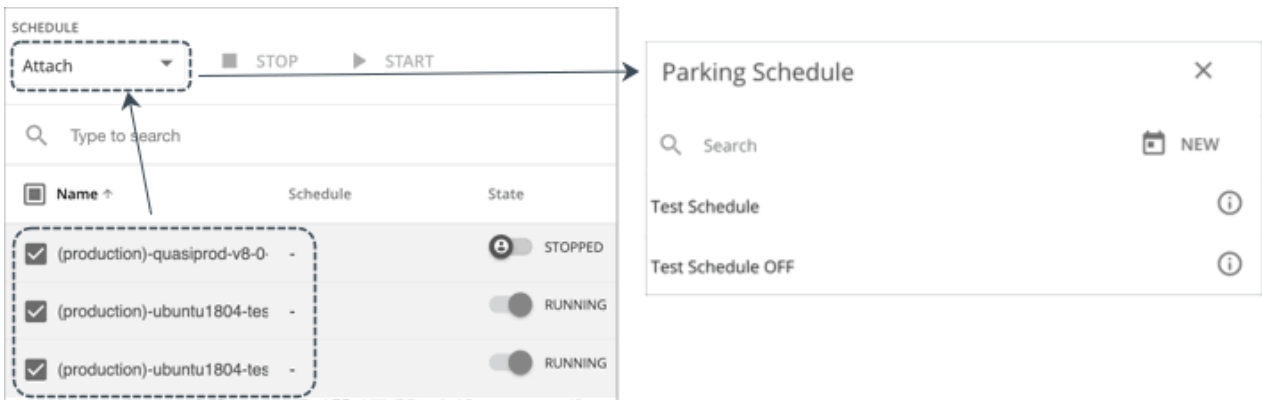
## Scheduling Parking Actions

Attach a parking schedule to entities to ensure that parking actions occur automatically at the appropriate times.

### Note:

Currently, parking actions cannot be scheduled using [automation policies](#) for cloud VMs and Database Servers. Automation policies (and the schedules configured in these policies) only apply to scale actions.

To attach a schedule, select one or several entities without a schedule and then select **Attach** from the dropdown at the top-left section of the page. In the Parking Schedule page, choose from a list of previously configured schedules.



If no schedule is suitable or available, create one by clicking **New** at the top-right section of the Parking Schedule page. Be aware that a schedule created this way will immediately apply to the entities in scope. If this is not your preference, create a new schedule by navigating to **Settings > Schedule** and then clicking **New Schedule > Parking Schedule**. For instructions on creating parking schedules, see [Managing Parking Schedules](#).

After attaching a schedule, the name of the schedule appears under the **Schedule** column. Click the name to view the settings for that schedule. To edit the settings, navigate to **Settings > Schedule** and then open the schedule for editing.

In the All Actions chart, the tabular view shows the following information for scheduled parking actions that have been executed:

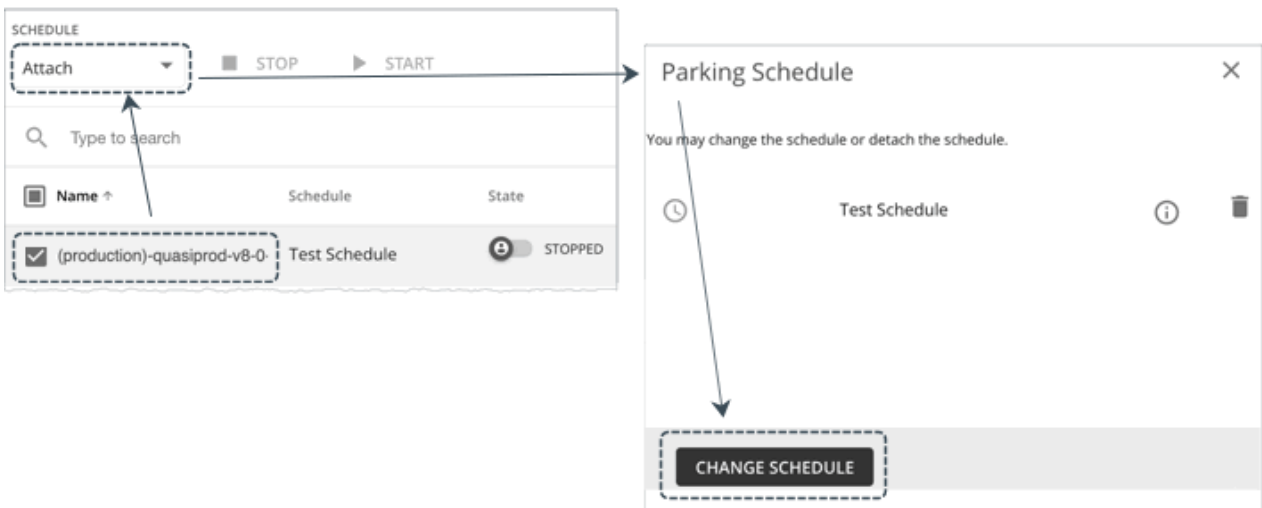


| All Actions          |                                               |                        |                                            |                      |
|----------------------|-----------------------------------------------|------------------------|--------------------------------------------|----------------------|
| mod-proxy - Day      |                                               |                        |                                            |                      |
| DATE CREATED         | ACTION DESCRIPTION                            | RISK TYPE              | EXECUTION                                  | DATE EXECUTED        |
| 02 Mar 2023 01:17 AM | Start Virtual Machine mod-proxy in [REDACTED] | Efficiency Improvement | Automatically Accepted By Test Schedule ON | 02 Mar 2023 01:17 AM |

- Execution: Automatically Accepted
- Risk Type: Savings (stop actions) or Efficiency Improvement (start actions)

## Changing a Parking Schedule

To change the schedule currently attached to an entity, select the entity and then select **Attach** from the dropdown at the top-left section of the page. In the Parking Schedule page, click **Change Schedule** and then choose from a list of previously configured schedules (or create a new one). When you are done, the new schedule displays under the **Schedule** column.



You can select multiple entities, as long as those entities share the same schedule.

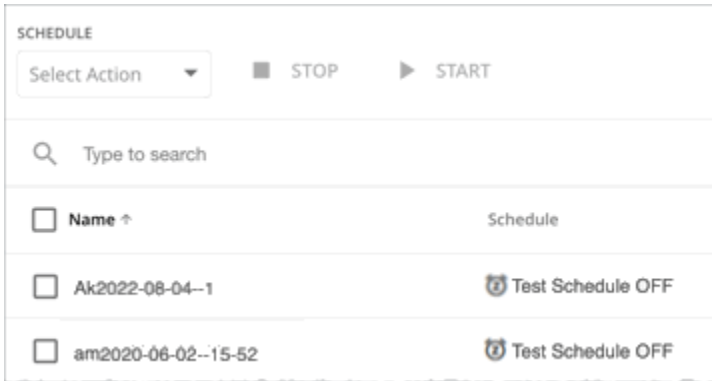
## Pausing a Parking Schedule

You can pause schedule enforcement for a given period of time. For example, you can pause during non-typical work hours or when an unanticipated maintenance window requires automation outside Turbonomic to start and stop workloads. If the entity was running, it keeps running. If it was stopped, it stays stopped.

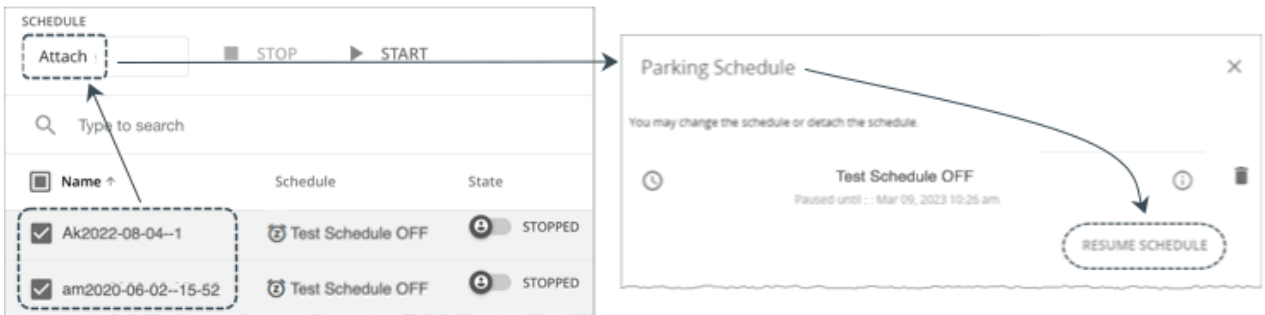
To pause a parking schedule, select one or several entities with a schedule, and then select **Pause** from the dropdown at the top-left section of the page. In the Pause Schedule page, specify the duration of the pause.



An icon appears before the name of the schedule to indicate that it has been paused.



The parking schedule will resume automatically when the pause period is over. To manually resume the schedule, select the affected entities, and then select **Attach** from the dropdown at the top-left section of the page. In the Parking Schedule page, click **Resume Schedule**.

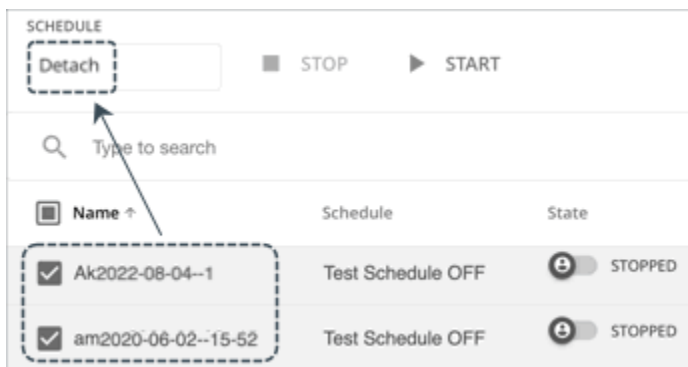


**Note:**

When you execute a parking action on demand and the affected entity has an existing schedule, a system prompt notifies you to pause the schedule before executing the action. This prevents potential conflicts between on-demand and scheduled parking actions.

**Detaching a Parking Schedule**

You can detach parking schedules anytime. Select one or several entities with schedules, and then select **Detach** from the dropdown at the top-left section of the page.



## General Tasks

- For a long list of entities, you can type a search keyword or add filters to narrow the results.
- To set an entity as your scope, click the entity name.
- To download the data in the Parking page, click the download button at the top-right section of the page.

## Place: Reserve Workload Resources

From the Workload Placement Page, you can set up reservations to save the resources you will need to deploy VMs at a future date. Turbonomic calculates optimal placement for these VMs and then reserves the host and storage resources that they need.

To reserve VMs, you will need to choose a VM template, specify any placement constraints, set how many instances to reserve, and then indicate whether to reserve now or in the future. Because reserved VMs do not yet exist, they do not participate in the real-time market.

### About VM Templates for Reservations

VM templates specify the resource requirements for each reserved VM, including:

- Compute and storage resources allocated to each VM
- Consumed factor. This is the percentage of allocated CPU, memory, or storage that the reserved VM will utilize.

For more information about these templates, see [VM Template Settings](#).

### About Placement of Reserved VMs

To determine the best placement for the VMs you want to reserve, Turbonomic runs a plan using the last-generated data in nightly-run headroom plans.

#### Note:

If you have changed your environment by adding targets or changing policies, wait until the next run of headroom plans for the affected scope before you create reservations.

When making placement decisions, Turbonomic considers the following:

- Placement constraints set in the reservation
- Demand capacity

Turbonomic calculates demand based on the *resource allocation* and *consumed factor* set in VM templates. For example, if you want to create a reserved VM from a template that assigns 3 GB of virtual memory and a consumed factor of 50%, Turbonomic calculates 1.5 GB of demand capacity for the reservation.

- Overprovisioned capacity

For reserved VMs, this corresponds to the resource allocation set in VM templates. Continuing from the previous example, Turbonomic assumes 3 GB of overprovisioned capacity for a reserved VM created from a template that assigns 3 GB of virtual memory.

For providers (hosts and storage), Turbonomic calculates overprovisioned capacity. The default overprovisioned capacity is 1000% for host Mem and CPU, and 200% for storage. A host with 512 GB of memory will have an overprovisioned capacity of 5 TB (5120 GB).

Providers must have sufficient *demand* and *overprovisioned* capacity to place a reservation. Turbonomic analyzes the current and historical utilization of cluster, host, and storage resources to identify viable providers for the VMs when they are deployed to your on-prem environment. In this way, Turbonomic can prevent congestion issues after you deploy the VMs.

**Note:**

Turbonomic persists historical utilization data in its database so it can continue to calculate placements accurately when market analysis restarts.

The initial placement attempt either succeeds or fails.

- Successful Initial Placements

If the initial placement attempt is successful, Turbonomic adds the reserved VM to your inventory.

In the previous example, a reserved VM that requires 1.5 GB of demand capacity and 3 GB of overprovisioned capacity can be placed on a host with 512 GB of memory (5 TB of overprovisioned capacity), assuming no constraints will prevent the placement.

Note that *actual* and *reserved* VMs share the same resources on providers. This means that provider capacity will change as demand from the actual VMs changes. Turbonomic polls your environment once per day to identify changes in provider capacity. It then evaluates if it can continue to place the reserved VMs *within the same cluster*, and then shows the latest placement status.

For example, if the host for a reserved VM is congested at the time of polling, Turbonomic might decide to move the VM to another host in the cluster that has sufficient capacity. In this case, the placement status stays the same (**Reserved**). Should you decide to deploy the VM at that point, you need to deploy it to the new host. If, on the other hand, there is no longer a suitable host in the cluster, the placement fails and the status changes to **Placement Failed**. Deploying the VM at that point will result in congestion. Turbonomic will *not* retry fulfilling the reservation.

- Failed Initial Placements

If the initial placement attempt is unsuccessful (for example, if all providers have seen historical congestion), Turbonomic shows that the placement has failed and will *not* retry fulfilling the reservation.

## Current and Future Reservations

You can create a current or future reservation from the Workload Placement Page.

- Current Reservation

Turbonomic calculates placement immediately and then adds the reserved VMs to your inventory if placement is successful.

This reservation stays in effect for 24 hours, or until you delete it.

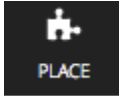
- Future Reservation

Set the reservation for some time in the future.

Turbonomic does not calculate placement at this time — the future reservation saves the definition, and Turbonomic will calculate placement at the time of the reservation start date.

This reservation stays in effect for the duration that you set, or until you delete it.

## Displaying the Workload Placement Page



To see the reservations that are currently in effect and to create new reservations, click the **PLACE** button in the Navigation Menu.

ALL RESERVATIONS

**Workload Placement**  
Workload Placement

**CREATE RESERVATION**

🔍 search... FILTER

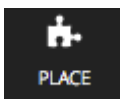
|                          |                                                                                  |                  |
|--------------------------|----------------------------------------------------------------------------------|------------------|
| <input type="checkbox"/> | 5 Reservations                                                                   |                  |
| <input type="checkbox"/> | <b>Cud_GiantVM</b><br>Cud_GiantVM 2/7/2020 - 3/7/2020 PLACEMENT_FAILED "GiantVM" | PLACEMENT F... > |
| <input type="checkbox"/> | <b>CudMultipleVMs</b><br>10 "Hatice_VM" placed on HawthorneDev                   | RESERVED >       |
| <input type="checkbox"/> | <b>CudRes4</b><br>2 "Hatice_VM" placed on HawthorneDev                           | RESERVED >       |
| <input type="checkbox"/> | <b>CudRes5</b><br>1 "Hatice_VM" placed on HawthorneDev                           | RESERVED >       |
| <input type="checkbox"/> | <b>MyReservation</b><br>MyReservation 2/7/2020 - 3/7/2020 RESERVED "Hatice_VM"   | RESERVED >       |

## Creating a Reservation

Reservations set aside resources for anticipated workload. While a reservation is in the RESERVED state, Turbonomic continually calculates placement for the reserved VMs.

To create a reservation:

1. Navigate to the Workload Placement page.



2. Create a new reservation.



In the Workload Placement page, click **CREATE RESERVATION**.

Turbonomic displays a list of templates. Choose the template you want, and click **NEXT: CONSTRAINTS**.

3. Optionally, specify placement constraints.

In the **Constraints** section and choose which constraints to apply to this reservation.

Constraints are optional, but note that these constraints are how you ensure that the template you have chosen is viable in the given locations that Turbonomic will choose.

The constraints you can choose include:

- Scope

Choose the datacenter or host cluster that you will limit the reservation to.

- Placement Policy

This list shows all the placement policies have been created as **Turbonomic Segments**. Choose which placement policies the reservation will respect.

- Networks

Turbonomic discovers the different networks in your environment. Use this constraint to limit workload placement to the networks you choose.

When you are done setting constraints, click **NEXT: RESERVATION SETTINGS**.

#### 4. Make the reservation settings, and create the reservation.

To finalize the reservation, make these settings:

- RESERVATION NAME

The name for the reservation. You should use unique names for all your current reservations.

This name also determines the names of the reservation VMs that Turbonomic creates to reserve resources in your environment. For example, assume the name *MyReservation*. If you reserve three VMs, then Turbonomic creates three reservation VMs named *MyReservation\_0*, *MyReservation\_1*, and *MyReservation\_2*.

- VIRTUL MACHINES COUNT

How many VMs to reserve.

**Note:**

You can include up to 100 VMs in a single reservation.

- RESERVATION DATE

The time period that you want the reservation to be active. Can be one of:

- Reserve Now

Use this to calculate the ideal placement for a workload that you want to deploy today.

Turbonomic begins planning the reservation immediately when you click **CREATE RESERVATION**. The reservation stays in effect for 24 hours – At that time Turbonomic deletes the reservation.

- Future Reservation

This executes the reservation for the date range you specify. Turbonomic begins planning the reservation on the day you set for START DATE. The END DATE determines when the reservation is no longer valid. At that time, Turbonomic deletes the reservation.

When you are finished with the reservation settings, click **CREATE RESERVATION**. Turbonomic displays the new reservation in the Workload Placement page. Depending on the reservation settings and your environment, the reservation can be in one of the one of the following states:

- UNFULFILLED

The reservation request is in the queue, waiting for an ongoing reservation request to complete.

- INPROGRESS

Turbonomic is planning the placement of the reservation workloads.

- FUTURE

Turbonomic is waiting for the START DATE before it will start to plan the reservation.

- RESERVED

Turbonomic has planned the reservation, and it found providers for all the VMs in the reservation.

As your environment changes, Turbonomic continues to calculate the placement for the reservation

VMs. If at any time it finds that it cannot place all the VMs, it changes the reservation to PLACEMENT FAILED.

- **PLACEMENT FAILED**

Turbonomic cannot place all the reservation VMs. As your environment changes, Turbonomic continues to calculate placement for the VMs. If at any time it finds that it can place all the VMs, it changes the reservation to RESERVED.

- **INVALID**

An error occurred while planning the placement of the reservation VMs.

**Note:**

The list of reservations refreshes whenever you open the Workload Placement page. To see changes in reservation state, navigate away from the page, and navigate back to it again.

## Managing Reservations

Click the name to open the Reservation Settings fly-out

Expand the list entry for details

| Host                     |  | Storage           |  |
|--------------------------|--|-------------------|--|
| hp-dl565.eng.vmturbo.com |  | QSGRID01:NFSShare |  |

| Reservation Name                                                                                   | State            | Details |
|----------------------------------------------------------------------------------------------------|------------------|---------|
| 5 Reservations                                                                                     |                  |         |
| <input type="checkbox"/> Cud_GiantVM<br>Cud_GiantVM 2/7/2020 - 3/7/2020 PLACEMENT_FAILED "GiantVM" | PLACEMENT FAILED | >       |
| <input type="checkbox"/> CudMultipleVMs<br>10 "Hatice_VM" placed on HawthorneDev                   | RESERVED         | ⌵       |
| <input type="checkbox"/> CudRes4                                                                   | FUTURE           | >       |
| <input type="checkbox"/> CudRes5<br>1 "Hatice_VM" placed on HawthorneDev                           | RESERVED         | >       |
| <input type="checkbox"/> MyReservation<br>MyReservation 2/7/2020 - 3/7/2020 RESERVED "Hatice_VM"   | RESERVED         | >       |

Click a provider name to drill down to that entity

The PLACE page displays the current list of reservations. You can expand items in the list to see some details, or you can click to view the full details. You can also select items to delete them, which cancels the reservation or deployment.

For an entry in the RESERVED state, you can click the entry name to open the Reservation Settings fly-out.

To delete a reservation, select it in the list and click the DELETE icon.

To see details about the provider entities, or the datacenter that is hosting the reserved VMs, click that entity name.

## Deploying Workloads to the Reserved Resources

When you reserve resources, you know that they will be available for you to deploy actual VMs in your environment. To deploy these VMs, you should:

1. Note the placement that your reservation has calculated.

Expand the reservation entry in the Workload Placement page and note the hosts and storage that will provide resources for your VMs.

2. Delete the reservation.

Before you deploy the reserved VMs, you should delete the reservation. This frees up the Turbonomic market to manage the placement of the VMs you are about to deploy.

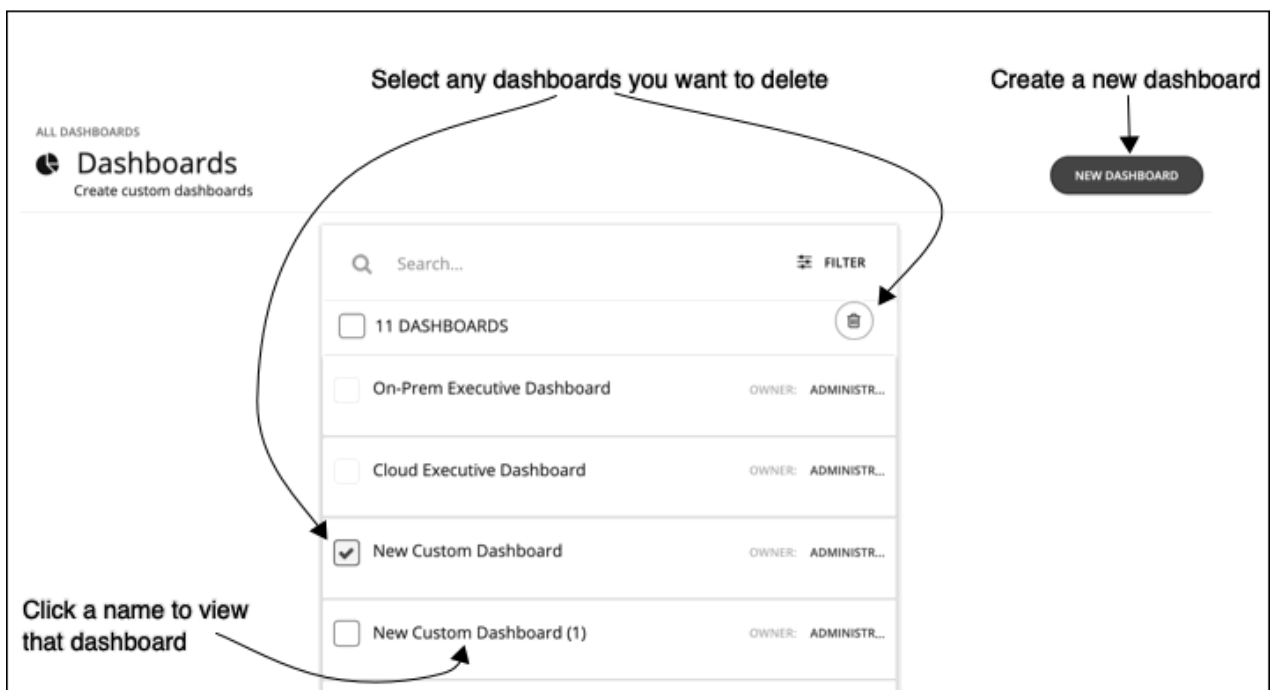
**Note:**

When you delete a reservation from the user interface or API, Turbonomic only marks the reservation for deletion and waits 48 hours before permanently deleting it. You can permanently delete a reservation by using the API's `reservation_force_delete` parameter along with a DELETE call to a specific reservation. When `reservation_force_delete = true`, the system removes the reservation permanently, no matter what state it is in.

3. Deploy the actual VMs.

In your Hypervisor user interface, deploy the VMs to the hosts and storage that you noted. When you are done, Turbonomic will manage their placement the same as it manages the rest of your environment.

## Dashboards: Focused Views



Dashboards give you views of your environment that focus on different aspects of the environment's health. At a glance, you can gain insights into service performance health, workload improvements over time, actions performed and risks avoided, and savings in cost. For cloud environments, you can see utilization of discounts, potential savings, required investments, and the cost/performance of specific cloud accounts.

The Dashboards page lists all the dashboards that are available to you, including built-in and custom dashboards that your account can access. To view a dashboard, click its name in the list.

Built-in dashboards give you overviews of your on-prem, cloud, and container environments, showing how you have improved your environment over time.

From the Dashboard page, you can also create your own custom dashboards.



**Note:**

In charts that show tables, if the table contains more than 500 cells, then the User Interface disables the option to export the chart as PDF. You can still export the chart as a CSV file to load in a spreadsheet.

## Built-in Dashboards

Built-in Dashboards are scorecards of your environment. They demonstrate how well you are improving performance, cost, and compliance, as well as opportunities for further improvements that are available.

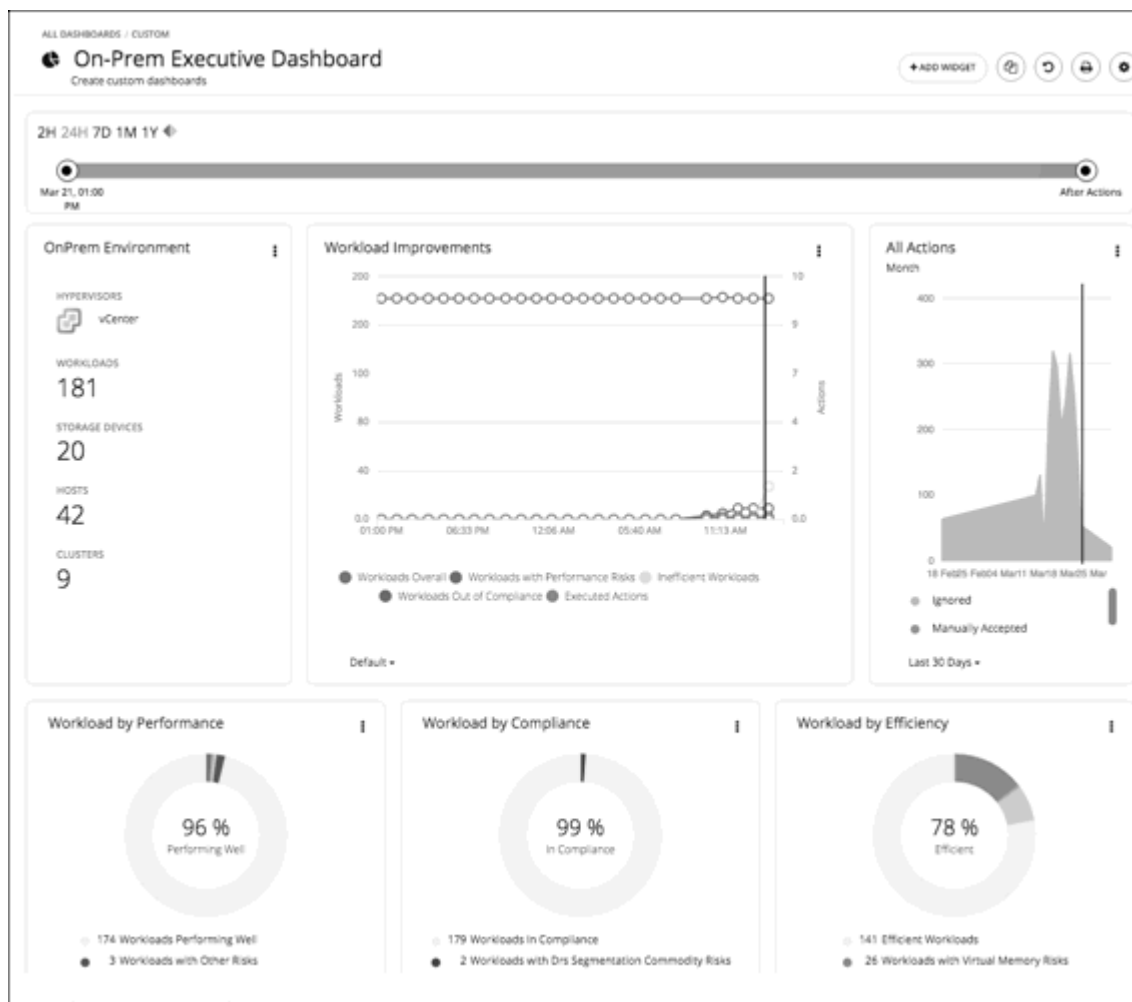
Turbonomic ships with these dashboards:

- On-Prem Executive Dashboard
- Cloud Executive Dashboard
- Container Platform Dashboard

**Note:**

Turbonomic ships these dashboards with default configurations. To edit a dashboard, you must log in with the administrator user account. Users logged in with that account can add or remove chart widgets, and change widget scopes. For information about editing dashboards, see [Creating and Editing Custom Dashboards](#).

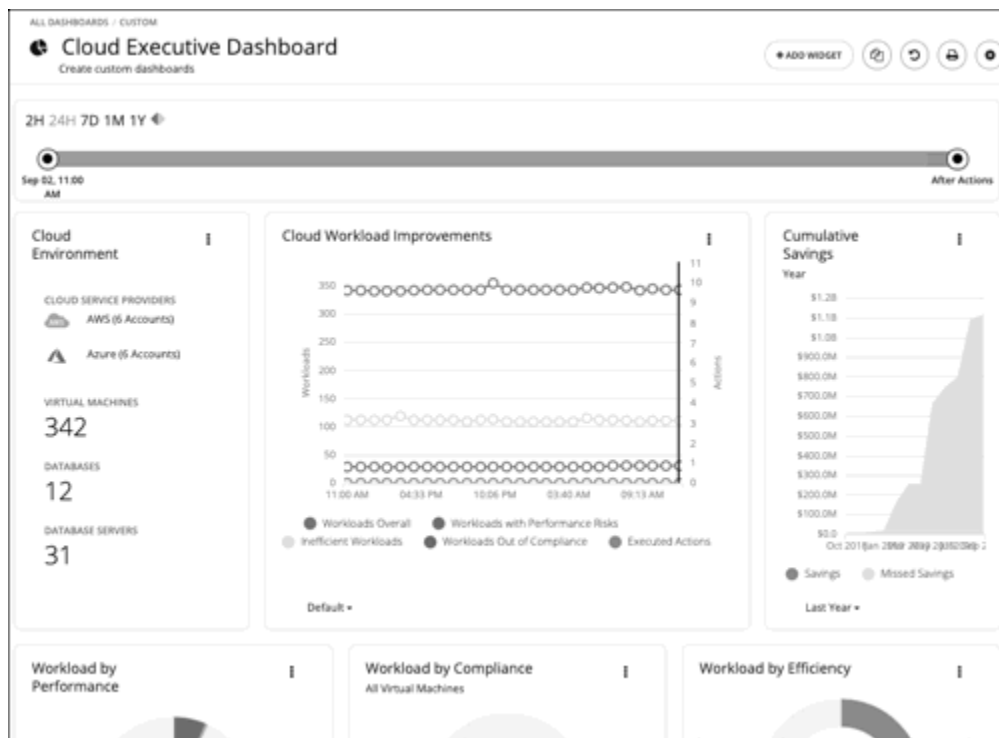
## On-Prem Executive Dashboard



The On-Prem Executive Dashboard shows the overall performance, capacity, and compliance in your on-prem infrastructure. This includes insights into:

- Actions History
  - The **On-Prem Environment** chart widget shows you an overview of your on-prem environment that Turbonomic is managing and controlling. The chart displays the workloads and the infrastructure that Turbonomic discovered.
  - The **Workload Improvements** chart widget shows how the efficiency, performance, and policy risks associated with your workloads have disappeared as you have increased your adoption of Turbonomic Workload Automation. The chart tracks how your workloads have grown as your execution of actions have increased or decreased as your environment achieves and maintains its desired states over time.
  - The **All Actions** chart widget shows the number of actions that Turbonomic has generated versus the ones executed. This gives you an understanding of where there were more opportunities for improvement that were not taken in the past versus those that are available today.
- Opportunities
  - The **Workload by Performance, Workload by Compliance, and Workload by Efficiency** chart widgets indicate workload health by showing the risks that are currently in your environment and each classification of those risks. You can click **Show Action** on the chart to reveal all of the outstanding actions that need to be taken to resolve those risks on your workloads.
  - The **Necessary Investments and Potential Savings** chart widgets together project how the current actions to improve performance, efficiency, and compliance will impact your costs.
- Current State
  - This chart shows the top clusters in your on-prem environment by CPU, memory, and storage capacity or utilization. In the default view, the chart shows the top clusters by CPU headroom (available capacity). It also shows time to exhaustion of cluster resources, which is useful for future planning (for example, you might need to buy more hardware).
  - The **Virtual Machines vs Hosts and Storage** and the **Virtual Machines vs Hosts and Storage -Density** chart widgets show how your overall density has improved in your on-prem environment. A high count of VMs per host or storage means that your workloads are densely packed.

## Cloud Executive Dashboard



The Cloud Executive Dashboard shows your overall cloud expenditures and how you can improve performance and reduce cost. This includes insights into:

- Actions History
  - The **Cloud Environment** chart widget shows you an overview of your cloud environment that Turbonomic is managing and controlling. The chart displays the workloads, cloud service providers, and cloud accounts that you currently have set up as Turbonomic targets.
  - The **Workload Improvements** chart widget shows how the efficiency, performance, and policy risks associated with your workloads have disappeared as you have increased your adoption of Turbonomic Workload Automation. The chart tracks how your workloads have grown as your execution of actions have increased or decreased as your environment achieves and maintains its desired states over time.
  - The **Cumulative Savings** chart widget shows you the cost savings for executed cloud actions compared to the cloud actions that you have not executed (missed savings).
- Opportunities
  - The **Workload by Performance**, **Workload by Compliance**, and **Workload by Efficiency** chart widgets indicate workload health by showing the risks that are currently in your environment and each classification of those risks. You can click **Show Action** on the chart to reveal all of the outstanding actions that need to be taken to resolve those risks on your workloads.
  - The **Necessary Investments** and **Potential Savings** chart widgets together project how the current actions to improve performance, efficiency, and compliance will impact your costs.
  - **Cloud Estimated Cost** chart widget shows estimated monthly costs and investments for the cloud. Monthly cost amounts are summarized as amounts with and without actions.
- Current State
  - The **Top Accounts** chart widget shows all of the cloud accounts in your cloud environment and what the utilization is for each account. You can see the number of workloads, estimated monthly costs, saved by actions, and actions taken. In the default view, the chart shows the top cloud accounts and you can click **Show All** button to see all of the accounts. In the Show All list, you can also download the account cost data as a CSV file or PDF.
  - The **Cost Breakdown by Tag** chart widget shows the tags you have assigned to your cloud resources and the costs associated with each of these tagged categories. The **Cost Breakdown by Cloud Service Provider** chart widget is an Expenses chart widget that shows your expenses for each cloud service provider.
  - Usage of Discounts

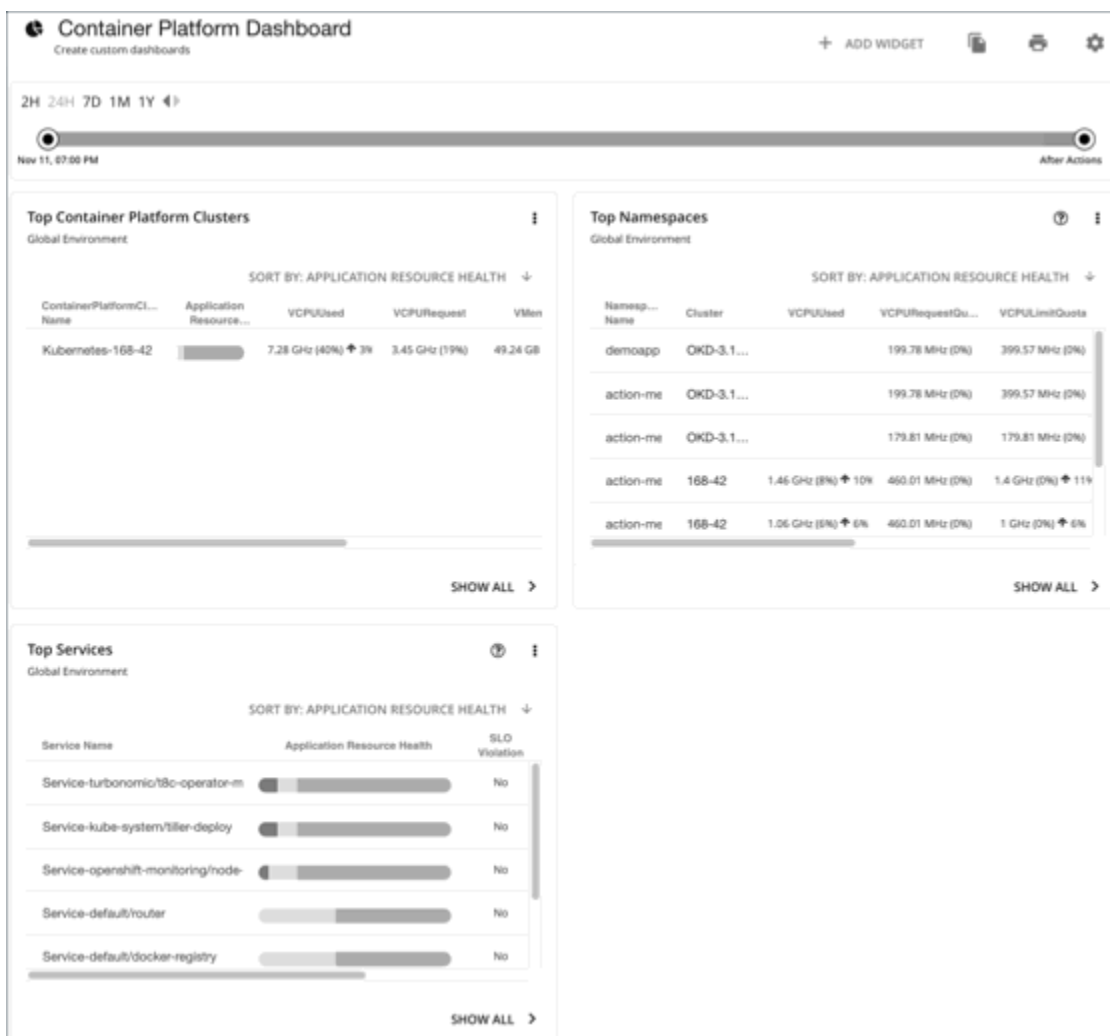
Discounts reduce cost by offering a subscription-based payment plan. Turbonomic discovers these discounts and tracks usage patterns to identify workloads that can take advantage of discounted pricing. The Cloud Executive Dashboard shows whether you are getting the most out of your current discounts.

    - Discount Coverage

This chart shows the percentage of VMs covered by discounts. If you have a high percentage of on-demand VMs, you should be able to reduce your monthly costs by increasing coverage. To increase coverage, you scale VMs to instance types that have existing capacity.
    - Discount Inventory

This chart lists the cloud provider discounts discovered in your environment.

## Container Platform Dashboard



The Container Platform Dashboard shows the overall performance, capacity, and health of your container infrastructure. This includes insights into:

- Top Container Platform Clusters  
Assess the health of your clusters and sort them by risk level.
- Top Namespaces  
Identify namespaces that are running out of quota, and how much resources each namespace is using in both quotas and actual utilization.
- Top Services  
Assess the impact of Services on the performance of your applications.

## Creating and Editing Custom Dashboards

A custom dashboard is a view that you create to focus on specific aspects of your environment. You can create dashboards that are private to your user account, or dashboards that are visible to any user who logs into your Turbonomic deployment.

Two common approaches exist for creating custom dashboards:

- Scope First  
You can create a dashboard in which all of the chart widgets focus on the same scope of your environment. For example, you might want to create a dashboard that focuses on costs for a single

public cloud account. In that case, as you add chart widgets to the dashboard, you give them all the same scope.

- Data First

You might be interested in a single type of data for all groups of entities in your environment. For example, each chart widget in the dashboard can focus on Cost Breakdown by Cloud Service, but you set the scope of each chart widget to a different cloud region or zone.

Of course, you can mix and match, according to your needs. You can set any scopes or data sources to the chart widgets in a dashboard to set up whatever organization and focus that you want.

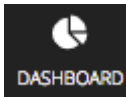
**Note:**

If you set a scope to your Turbonomic session, the specified scope does not affect your custom dashboards. For information about scoped views, see [Working With a Scoped View](#).

## Creating a Dashboard

To create a custom dashboard:

1. Navigate to the Dashboards Page.



Click to navigate to the Dashboard Page.

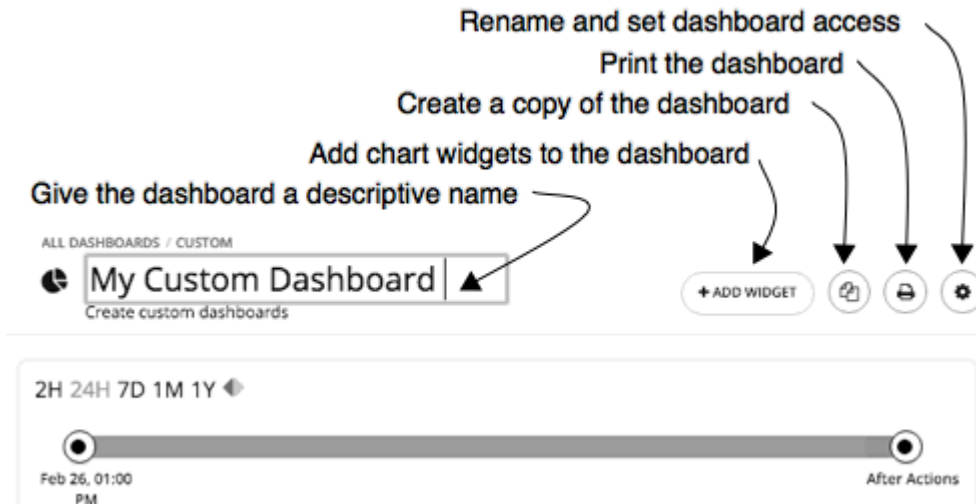
This page lists all dashboards that are available to you.

To view a dashboard, click its name in the list.

2. Create a new dashboard.



Click **NEW DASHBOARD** to add a new dashboard to your Turbonomic session. The dashboard appears with a default name and without chart widgets. The time range in the Time Slider is set to 24 hours by default.



3. Name the dashboard.

Give a name that describes the dashboard. If you will share the dashboard with all Turbonomic users, the name will help them decide whether to view it.

4. Add chart widgets to the dashboard.



Add as many chart widgets to the dashboard as you want. See [Creating and Editing Chart Widgets](#).

5. Optionally, set the dashboard access.

Click **Gear** to change the setting.

Dashboard access can be:

- **Only Me** – The dashboard is only available to your Turbonomic user account.
- **All Users** – Every Turbonomic user can see this dashboard.

By default, access is set to **Only Me**.

As soon as you create a new dashboard, it appears in the list on the Dashboard Page. Users with access to it can click the dashboard name in the list to view it.

At any time, if you are an administrator or the dashboard owner, you can view and make the following changes to the dashboard:

- Add, edit, or delete widgets
- Change the dashboard name
- Change the dashboard access setting

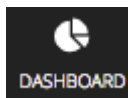
For executive dashboards, only an administrator (username=administrator) can edit an executive dashboard.

## Editing a Dashboard

If you have created a dashboard, you can change the name of the dashboard, its access settings, and its chart widgets. To change the chart widgets, see [Creating and Editing Chart Widgets](#).

To edit a dashboard's name or change its access settings:

1. Navigate to the Dashboards Page.



Click to navigate to the Dashboard Page.

2. Click the name of the dashboard that you want to edit.

3. Click **Gear** in the dashboard.

In the dashboard's Edit fly-out, make your changes.

Change the dashboard name

My New Dashboard Name

administrator

All Users

Set dashboard access

For the dashboard's access, you can set:

- **Only Me** – The dashboard is only available to your Turbonomic user account.
- **All Users** – Every Turbonomic user can see this dashboard.

4. When you are done, close the fly-out panel.

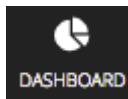
Your changes take effect when you close the fly-out.

## Deleting a Dashboard

If you are an administrator or the dashboard owner, you can delete a custom dashboard. You cannot delete executive dashboards.

To delete a custom dashboard:

1. Navigate to the Dashboards Page.



Click to navigate to the Dashboard Page.

This page lists all dashboards that are available to you.

2. Delete one or more dashboards.

In the list, choose the checkbox for each dashboard you want to delete and click **Trash can**.

## Creating and Editing Chart Widgets

Turbonomic displays information about your environment in various chart widgets. To focus on the information you need, you can add new chart widgets to scoped views and dashboards, and you can edit existing chart widgets. You can also pull the corners of chart widgets to resize them and change the display order of chart widgets in dashboards.

When you create or edit a chart widget, you can choose a variety of settings. For example, in the Top Utilized chart widget, if you choose Clusters as the Entity Type, you can then choose Utilization as the Data Type and Storage Provisioned as the Commodity.

### Creating a Chart Widget

To create a new chart widget:

1. Click **Add Widget** to open the Widget Gallery.

### In a scoped view

+ Add Widget

### On a dashboard

+ ADD WIDGET

On a dashboard, click **Add Widget** at the top-right corner. In a scoped view, click **Add Widget** on the right above the charts.

## 2. Choose a chart widget in the Widget Gallery.

The Widget Gallery is a list of thumbnail previews of chart widgets.

You can scroll through the gallery or search it. For example, if you type "Health" in the **Search** field, the results are two chart widgets, Health and Workload Health. You can choose chart widgets from these categories:

- Actions and Impact
- Status and Details
- Cloud
- On-Prem

To see the possible displays of a specific chart widget, use the horizontal scroll bar at the bottom of the thumbnail to scroll through the display choices.

To choose a chart widget to add it to your dashboard, click the thumbnail preview.

The Widget Preview window with the Edit fly-out opens.

## 3. Configure the settings for your chart widget.

Chart widget settings determine the data that the chart widget will show.

In the Edit fly-out, choose the settings and click **Update Preview** to display the result in the Widget Preview pane.

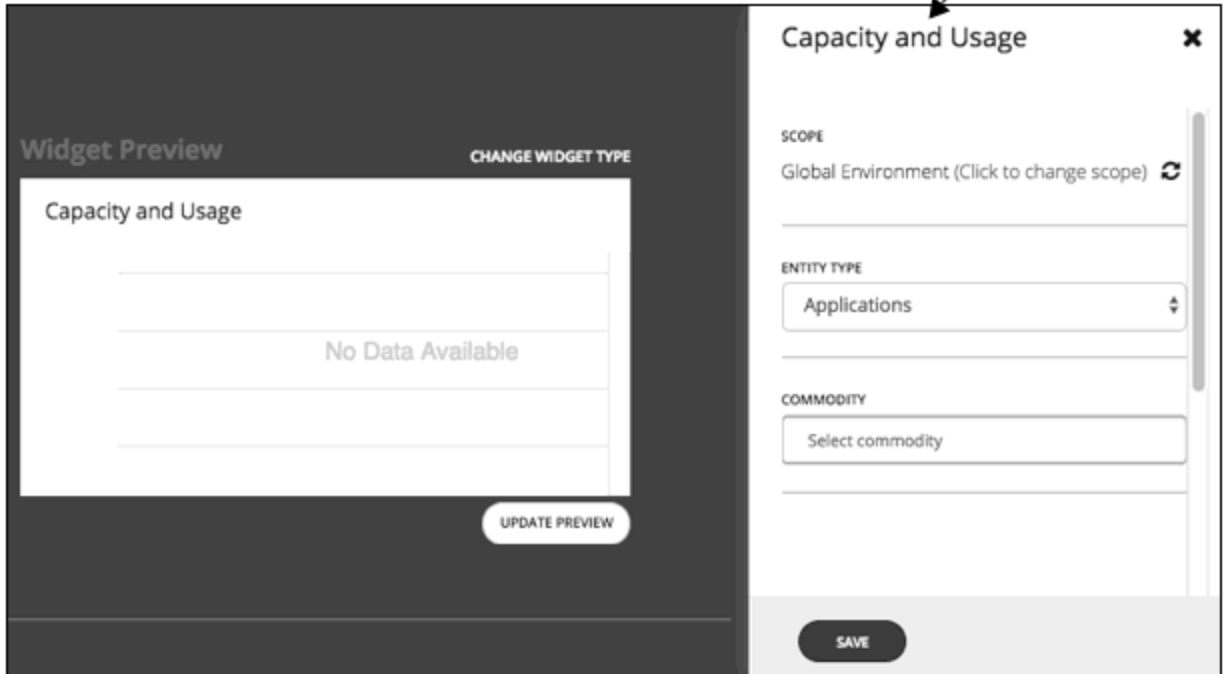
When you are satisfied with your settings, click **Save**. The chart widget is added to your dashboard.

For information about settings, see [Chart Widget Settings](#).

For example:



Choose the settings in the Edit fly-out

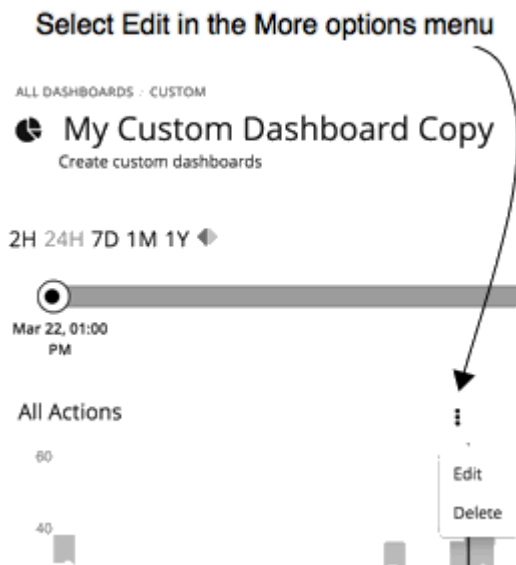


To delete a chart widget from your dashboard, choose **Delete** in the More options menu at the top-right corner of the chart widget.

## Methods to Access Chart Widget Settings

Two methods exist for accessing the chart widget settings in the Edit fly-out:

- You can access the settings in the Edit fly-out when you add a chart widget to your dashboard after you click a thumbnail preview.
- For an existing chart widget in a dashboard, you can choose **Edit** in the More options menu at the top-right corner.



## Chart Widget Settings

Chart widget settings vary according to the type of chart widget. Also, depending on the value that you choose for a setting, additional settings may appear. The following is a list of frequently-used chart widget settings:

- Scope

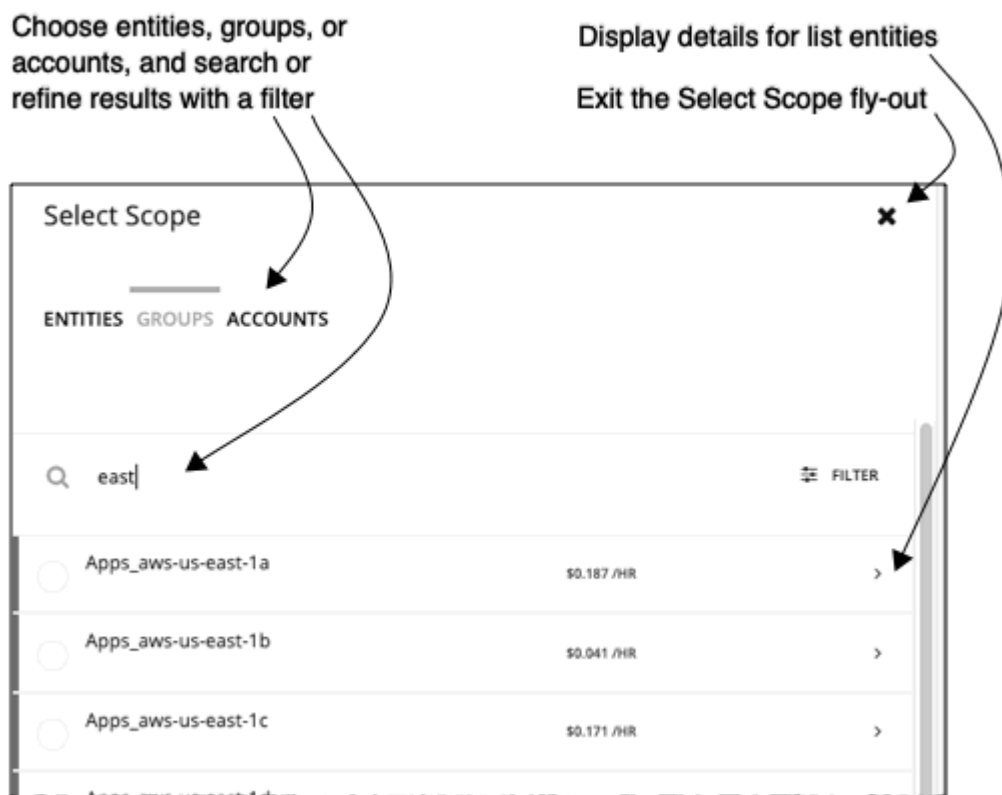
The set of entities in your environment that this chart widget represents. By default, the chart widget scope is set to **Global Environment**.

For every type of chart widget, you have the option to set the chart's scope. To do so:

1. Click **Click to change scope** to open the Select Scope fly-out.
2. In the Select Scope fly-out, choose the entity, group, or account that you want.

The ACCOUNTS tab is available depending on the type of chart widget.

Your choice appears in the **Scope** field.



- Timeframe

The timeframe for historical data or projections in the chart. Choices for the chart's timeframe are: Default, Last 2 Hours, Last 24 Hours, Last 7 Days, Last 30 Days, and Last Year.

If you set the timeframe to **Default**, the dashboard Time Slider controls the timeframe setting. For example, if your dashboard Time Slider is set to one month (1M), then all chart widgets with the Default timeframe in that dashboard are set to one month and show information for one month. Note that the dashboard Time Slider does not override the other specific timeframe settings.

- Chart Type

The chart widget's display type. Most chart widgets can display horizontal bar or ring charts. Other display choices can include tabular data, band chart, stacked bar, line, or area charts.

**Note:**

For summary charts like horizontal bar and ring charts, when the legend has more than four categories, the remaining categories are represented as a fifth category named "Other."

- Entity Type

The type of entities or their data that you want to display in this chart widget. Choices vary (for example, Applications, Hosts, Virtual Data Centers, Storage Devices, and so on).

- Commodity

The resources that you want this chart widget to monitor. Some charts can monitor multiple commodities. Choices vary (for example, CPU, Memory, Virtual Storage, and so on).

## Chart Types

Turbonomic provides many different types of charts in the Widget Gallery. To design dashboards, you should be familiar with the data each chart presents. These charts provide information on actions, impact, status of your environment, and details about specific entities, cloud, and on-prem environments.

### Actions and Impact Chart Types

These chart widgets provide information on actions, pending actions, risks that you avoided, improvements, and potential savings or investments.

#### *Pending Actions Charts*

Pending Actions charts show the actions that Turbonomic recommends to improve the current state of your environment.

### Chart Type

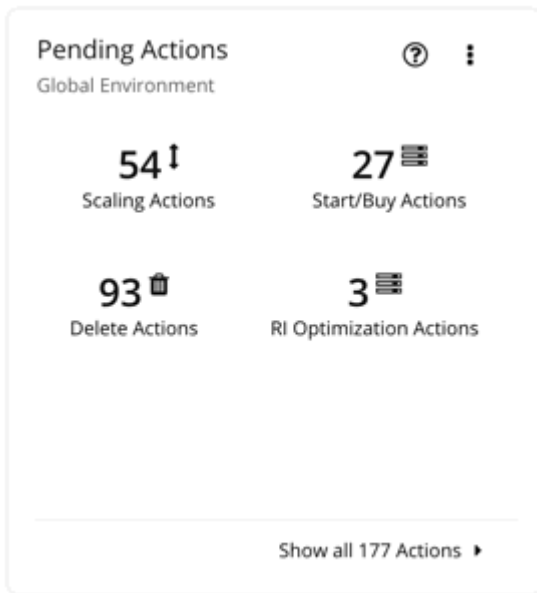
You can set the display to:

- Text
- Ring Chart
- Horizontal Bar
- List

Examples:

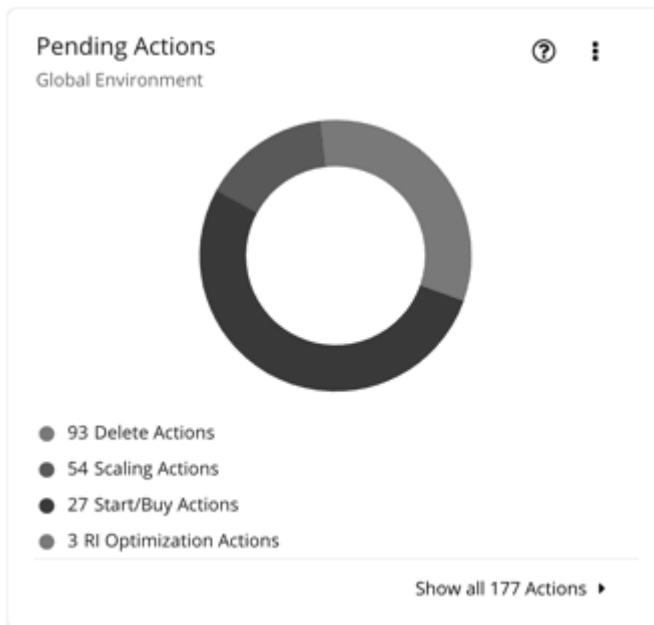
- Text

The text chart shows the number of actions for each action type. It gives a quick visual indication of the kinds of actions that are pending. For details, see [Action Types](#).



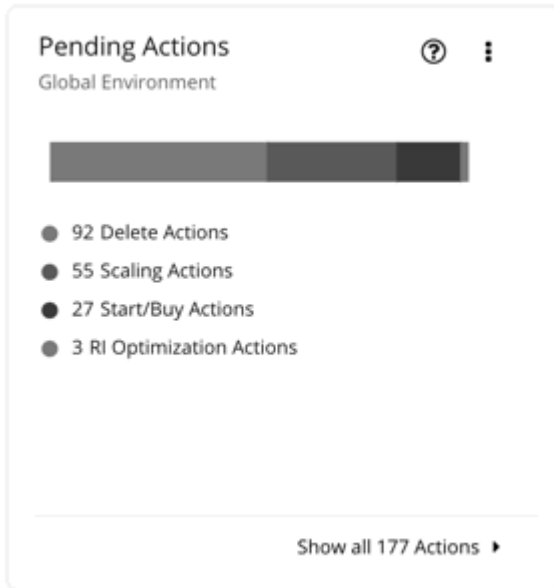
- Ring Chart

The ring chart counts the number of actions for each action type. It gives a quick visual indication of the kinds of actions that are pending. For details, see [Action Types](#).



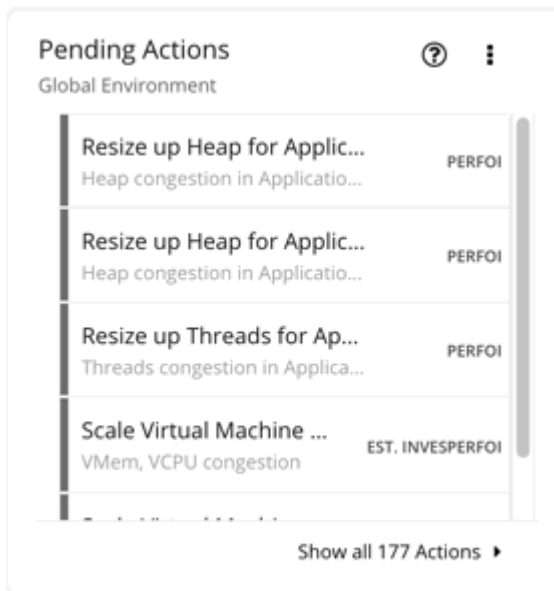
- Horizontal Bar

The horizontal bar chart counts the number of actions for each action type. It gives a quick visual indication of the kinds of actions that are pending. For details, see [Action Types](#)



- List

The list chart shows an abbreviated listing of the actions for the chart's scope. For details about the different actions generated by the product, see [Actions](#).



At the bottom of the chart, click **Show All** to see a full list of pending actions that are in the scope of the chart, along with action details and controls to execute actions. For details, see [Working with Action Center](#).

### **Actions Charts**

Actions charts keep a history of pending (not executed) and executed actions. These charts use historical data from the Turbonomic database. You can set the chart to show hourly, daily, or monthly data points.

### **Filter**

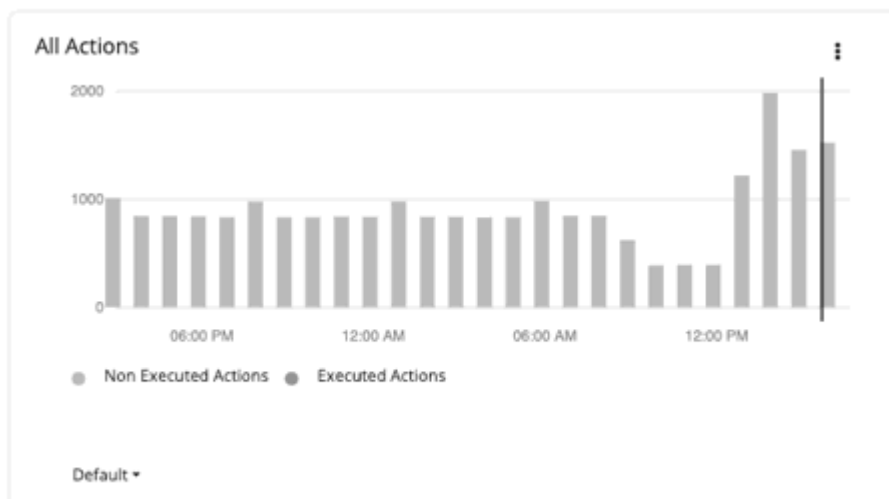
You can filter the chart to show **All Actions** (pending and executed actions) or only **Executed Actions**.

## Chart Type

You can set the display to:

- Tabular
- Area Chart
- Text
- Stacked Bar Chart

## Stacked Bar Chart



For the Stacked Bar Chart, each bar represents a time period. Hover over the bar to see the number of *unique* actions for that time period. In the default view, the bars represent actions per hour in the last 24 hours. The 2:00 PM bar, for example, shows actions between 2:00 PM and 2:59 PM.

A pending action that remains valid for an extended period of time is counted *once* for each hour, day, and month it remains pending. This also applies to pending actions that go away as conditions in the market change, but are generated again in the future. Once a pending action is executed, it is counted once (this time, as an executed action) on the hour, day, and month of execution.

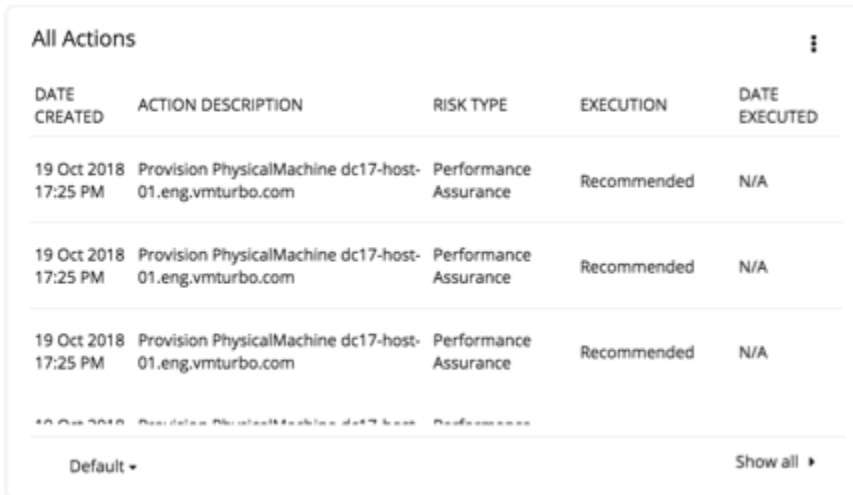
Consider the following scenarios.

- An action was generated at 1:25 PM and then executed two hours later at 3:25 PM.
  - For per-hour views (Last 24 Hours or Default), the action will be counted three times – as a *pending* action in the 1:00 PM and 2:00 PM bars, and as an *executed* action in the 3:00 PM bar.
  - For per-day (Last 7 or 30 Days) or per-month (Last Year) views, the action will be counted once (as an executed action) on the day or month of execution.
- An action was generated at 6:20 PM but went away (without being executed) in the next hour. The same action was generated again the next day at 9:10 AM and was executed immediately.
  - For per-hour views, the action will be counted twice – as a pending action in the 6:00 PM bar and as an executed action in next day's 9:00 AM bar.
  - For per-day views, the action will also be counted twice – as a pending action on Day 1 and an executed action on Day 2.
  - For per-month views, the action will be counted once (as an executed action) on the month of execution.

Use the chart to evaluate the rate of action execution, which underscores the importance of executing actions in a timely manner. As pending actions persist, they become more challenging to track and your environment stays in a risky state longer. To reduce potential delays in executing actions, consider action automation.

## Tabular Chart

To see the full list of actions, click **Show All** at the bottom of the chart.



| DATE CREATED         | ACTION DESCRIPTION                                     | RISK TYPE             | EXECUTION   | DATE EXECUTED |
|----------------------|--------------------------------------------------------|-----------------------|-------------|---------------|
| 19 Oct 2018 17:25 PM | Provision PhysicalMachine dc17-host-01.eng.vmturbo.com | Performance Assurance | Recommended | N/A           |
| 19 Oct 2018 17:25 PM | Provision PhysicalMachine dc17-host-01.eng.vmturbo.com | Performance Assurance | Recommended | N/A           |
| 19 Oct 2018 17:25 PM | Provision PhysicalMachine dc17-host-01.eng.vmturbo.com | Performance Assurance | Recommended | N/A           |

Default ▾ Show all ▶

## Risks Avoided Charts

As you execute the actions Turbonomic has recommended, you improve your environment's health and avoid risks to performance or cost. These charts show how many risks you have avoided over time. For example, the charts can show how many over-provisioning and congestion risks you avoided.

## Chart Type

You can set the display to:

- Text
- Ring Chart
- Horizontal Bar

## Optimized Improvements Charts

Turbonomic automatically executes or recommends actions, depending on the policies that you set up. For the recommended actions, you can use Optimized Improvements charts to show how utilization of resources would change assuming you accept all of the pending actions.

## Entity Type

Entity types you can choose include:

- Business Applications
- Business Transactions
- Services
- Application Components
- Chassis
- Containers
- Container Pods
- Container Specs
- Namespaces
- Workload Controllers
- Data Centers

- Databases
- Database Servers
- Disk Arrays
- IO Modules
- Internet
- Logical Pool
- Networks
- Hosts
- Regions
- Storage Devices
- Storage Controllers
- Switches
- Virtual Data Centers
- Virtual Machines
- Volumes
- Zones

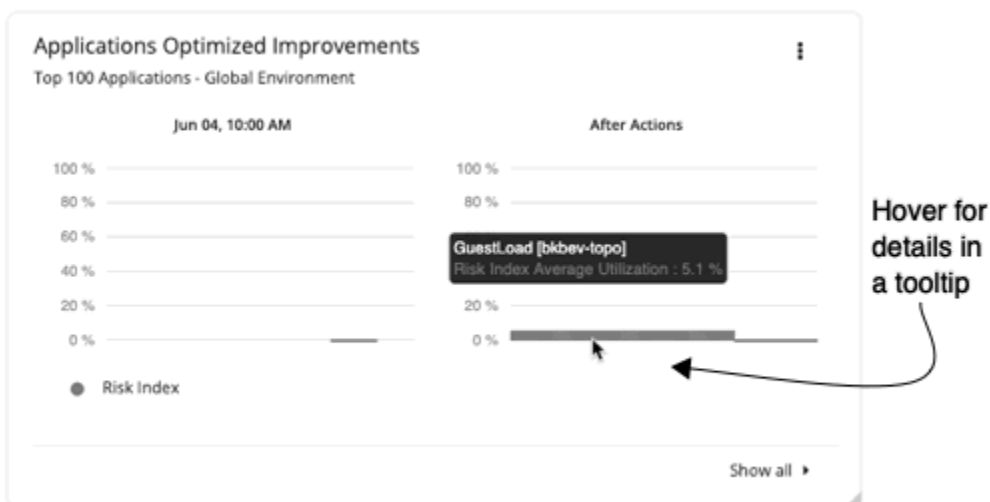
## Commodity

Depending on the entity type, you can add different resource commodities that you want to measure. For a chart of Hosts, you can measure commodities such as CPU, Memory, and even network flow between VMs that are on the same host (In-Provider Flow) or on other hosts (In-DPOD or Cross-DPOD Flow).

## Display

Optimized Improvements charts show two bar charts for the entities that are in scope – one for current consumption, and the other for the consumption you would expect to see if you execute all the actions.

Example: An Optimized Improvements chart for applications



## Potential Savings or Investments Charts

These charts show potential savings or necessary investments in your cloud expenditure, assuming you execute all the pending actions that Turbonomic identifies as a result of its analysis.



For example, if some workloads risk losing performance, Turbonomic might recommend scaling actions for the virtual machine to increase resources. The Necessary Investments chart shows how these actions translate to an increase in expenditure.

On the other hand, if there are pending actions to scale a virtual machine, which result in reduced monthly costs, the Potential Savings chart shows the reduced cost that would result from those actions.

This chart also track discount optimization actions. VM scaling actions may result in freed up capacity on a discounted instance type, which can now be applied to a different VM. Discount optimization actions reflect the potential savings resulting from reassigning the capacity to a different VM. These actions are not executed by Turbonomic users. They reflect capacity reassignment performed by your cloud provider.

The projected amounts include on-demand costs for VMs. For information about on-demand cost calculations, see [Estimated On-demand Monthly Costs for Cloud VMs](#).

## Type

You can choose **Potential Savings** or **Necessary Investments**.

## Chart Type

You can set the display to:

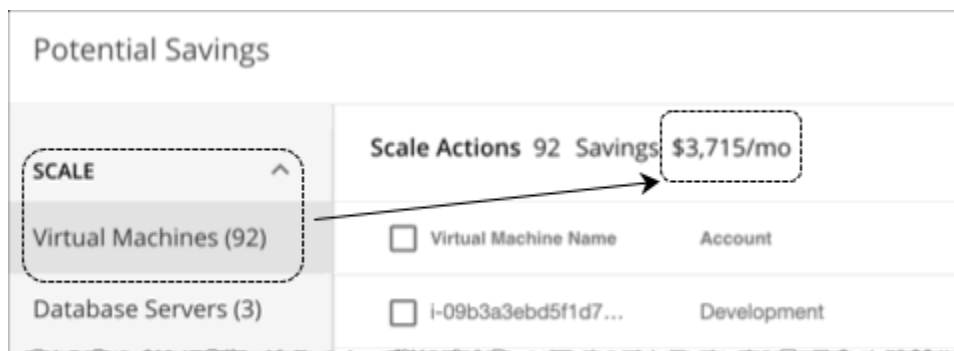
- Text
- Ring Chart
- Horizontal Bar

For the ring chart, you can click an action type (for example, **Scale Volumes**) in the chart or legend to display a filtered view of the actions list.

## Show All

Click **Show all** at the bottom of the chart to see a breakdown of savings or investments by action/entity type and entity. By default, the actions are shown in order of largest amounts so you can easily identify which ones will incur the highest costs or introduce the most savings.

For example, you can see the savings you would realize if you execute all *Scale* actions on the *virtual machines* included in the chart's scope.



The table then breaks down the total savings by individual virtual machines, and includes links to the specific actions that you need to perform to realize those savings.

The screenshot shows a 'Potential Savings' table with a filter set to 'SCALE'. The table displays 'Scale Actions 92 Savings \$3,715/mo'. The table has columns for 'Virtual Machine Name', 'Account', 'Instance Type', 'RI Cover...', 'On-Demand Cost', 'New Instance Type', 'New RI Coverage', 'New On-Dem...', 'Savings', and 'Action'. The table lists two rows of data for 'Database Servers (3)'.

| Virtual Machine Name | Account  | Instance Type | RI Cover... | On-Demand Cost | New Instance Type | New RI Coverage | New On-Dem... | Savings  | Action  |
|----------------------|----------|---------------|-------------|----------------|-------------------|-----------------|---------------|----------|---------|
| i-09b3a3ebd5f1d7...  | Dev      | t2.xlarge     | 0%          | \$0.371/hr     | m5.2xlarge        | 100%            | \$0/hr        | \$271/mo | DETAILS |
| eks-cluster-eks-w... | Advanced | t2.xlarge     | 0%          | \$0.371/hr     | m5.2xlarge        | 100%            | \$0/hr        | \$271/mo | DETAILS |

You can also compare instance types, costs, and discount coverage before and after executing the actions, allowing you to easily identify actions with the most savings.

## Status and Details Chart Types

These chart widgets provide information on the status of your environment and details about specific entities.

### *Health Charts*

Health charts show the current status of your environment, by entity type. For example, you can choose to show the health of all hosts in your environment, or the health of all the workloads running on a public cloud region.

### Entity Type

Entity types you can choose include:

- Business Applications
- Business Transactions
- Services
- Application Components
- Chassis
- Containers
- Container Pods
- Container Specs
- Namespaces
- Workload Controllers
- Data Centers
- Databases
- Database Servers
- Disk Arrays
- IO Modules
- Internet
- Logical Pool
- Networks
- Hosts
- Regions
- Storage Devices
- Storage Controllers
- Switches
- Virtual Data Centers
- Virtual Machines
- Volumes
- Zones

### Chart Type

You can set the display to:

- Text

- Ring Chart
- Horizontal Bar

### ***Basic Info Charts***

The Basic Info charts provide an overview of a single entity or individual Azure resource group that you have chosen as your scope.

#### **Type**

You can choose:

- **Entity Information**

This chart shows basic information (ID, Name, Type, State, Severity, Target Name, and so on) for the scoped entity or Azure resource group.

- **Related Tag Information**

This chart lists any available tag information for the scoped entity or Azure resource group. For example, in a cloud environment, if a virtual machine has tags applied to it, the chart shows those tags for the virtual machine.

#### **Display**

The chart shows the information as Tabular.

### ***Capacity and Usage Charts***

These charts list the resources for the selected entity type, showing their allocated capacity and how much of that capacity is in use.

#### **Entity Type**

Entity types you can choose include:

- Business Applications
- Business Transactions
- Services
- Application Components
- Containers
- Container Pods
- Container Specs
- Namespaces
- Workload Controllers
- Data Centers
- Database Servers
- Disk Arrays
- Logical Pool
- Networks
- Hosts
- Regions
- Zones
- Storage Devices
- Storage Controllers

- Virtual Machines
- Volumes

## Commodity

Depending on the entity type, you can add different resource commodities that you want to measure. For example, for a chart of Virtual Machines, you can measure commodities such as virtual CPU, memory, and storage.

### Note:

For a cloud database server, the chart might show incorrect *used* values for vMem and Storage Amount after an action executes. It could take up to 40 minutes for the correct values to display.

## Display

The chart shows the information as Tabular.

## Multiple Resources Charts

Multiple Resources charts show the historical utilization of commodities for the scoped entity or a group of entities. The vertical bar shows the current moment – plots that extend to the right project utilization into the future.

## Entity Type

Entity types you can choose include:

- Business Applications
- Business Transactions
- Services
- Application Components
- Containers
- Container Pods
- Container Specs
- Namespaces
- Workload Controllers
- Data Centers
- Database Servers
- Disk Arrays
- Logical Pool
- Networks
- Hosts
- Regions
- Zones
- Storage Devices
- Storage Controllers
- Virtual Machines
- Volumes

## Commodity

Depending on the entity type, you can add different resource commodities that you want to measure. For example, for a chart of volumes, you can measure commodities such as IO throughput, storage access, and storage amount.

## Show Peaks

Edit the chart and choose the **Show Peaks** checkbox to include peak information in the chart.

## Display

The chart shows the historical utilization and, if chosen, the peak information as a Line chart.

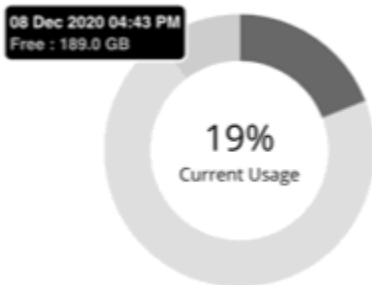
## Resources Charts

Resources charts show the utilization of a resource over time, for the entities in the chart's scope. The chart title shows the resource that you are plotting, as well as the chart's current scope.

To see finer details about your environment, you can set up charts that show utilization of specific commodities. For example, you can set up a dashboard with a number of Resources charts with their scopes set to the same cluster. Such a dashboard gives you a detailed look at the health of that cluster. Or you could make a dashboard with each chart scoped to a different cluster, but have all the charts show the same resource utilization.

## Ring Chart

For certain entity types (such as hosts, storage, and disk arrays), you will see a ring chart on the left that indicates the current overall utilization of a particular resource. Hover over the ring chart to see the following information:



- Free: Available capacity
- Used: Utilized capacity
- Reserved: Unavailable capacity due to utilization constraints

The sum of *Free* and *Used* capacity equals the total allocated capacity.

In addition to showing the currently discovered free and used capacity, Turbonomic also calculates *Reserved* capacity based on utilization constraints set in policies.

For example, for a cluster with 100 GB of allocated storage, Turbonomic might discover 80 GB of free capacity, and 20 GB of used capacity. If the cluster is currently applying a storage policy that has a utilization constraint of 90%, then Turbonomic will show 10 GB of reserved capacity.

## Options

Choose **Show Utilization** to see averages and peaks/lows, or **Show Capacity** to see averages and peaks/lows versus capacity.

The **Show Summary** option adds a ring chart to the view, showing the current utilization of the selected commodity.

## Chart Type

You can set the following types of display:

- Line Chart

A line plot showing resource utilization over time. The vertical green bar shows the current moment – Plots that extend to the right project utilization into the future.

- Band Chart

Lines plot average capacity and average used. The chart shows a band where its thickness indicates peaks and lows.

### Carbon Footprint Chart

Carbon footprint is the measurement of carbon dioxide equivalent (CO2e) emissions for a given entity. Turbonomic measures carbon footprint in grams.

Turbonomic collects energy-related data from hosts and VMs at 10-minute intervals, and then uses that data to calculate carbon footprint. Currently, Turbonomic collects data via vCenter, UCS Manager, and HPE OneView targets. When you set the scope to one or several hosts or VM discovered from these targets, the data that Turbonomic calculated displays in the Carbon Footprint chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

#### Note:

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

Turbonomic also relies on host power data for the supported targets. This data is mandatory for Energy and Carbon Footprint calculations.

## Supported Entities

Data is available in the Carbon Footprint chart for entities discovered via the following targets:

| Target Type | Target      | Supported Entities    |
|-------------|-------------|-----------------------|
| Fabric      | HPE OneView | Host, Virtual Machine |
|             | UCS Manager | Host                  |

| Target Type | Target  | Supported Entities    |
|-------------|---------|-----------------------|
| Hypervisor  | vCenter | Host, Virtual Machine |

## Carbon Footprint Calculation

The calculation for carbon footprint can be expressed as follows:

$$\text{Carbon Footprint} = (\text{Energy Consumption}) * [(\text{Power Usage Effectiveness}) * (\text{Carbon Intensity})]$$

Where:

- Energy Consumption is the consumption data collected from the entity.
- Power Usage Effectiveness (PUE) is a ratio that describes how efficiently a computer data center uses energy; specifically, how much energy is used by the computing equipment. PUE is the ratio of the total amount of energy used by a computer data center facility to the energy delivered to computing equipment. The closer PUE is to 1, the more efficient the computer data center.
- Carbon Intensity (CI) is a measurement of how 'clean' electricity is. It refers to how many grams of carbon dioxide (CO2) are released to produce 1 watt-hour (Wh) of electricity. Electricity that is generated using fossil fuels is more carbon intensive, as the process by which it is generated creates CO2 emissions. Renewable energy sources, such as wind, hydro, or solar power produce next to no CO2 emissions, so their carbon intensity value is much lower and often zero.

By default, PUE is set to 1.5, while CI is set to 0.25 g/Wh. These values appear in the [default policy](#) for Data Center entities. You can modify these values directly for a global effect, or set specific values in custom automation policies for Data Centers.

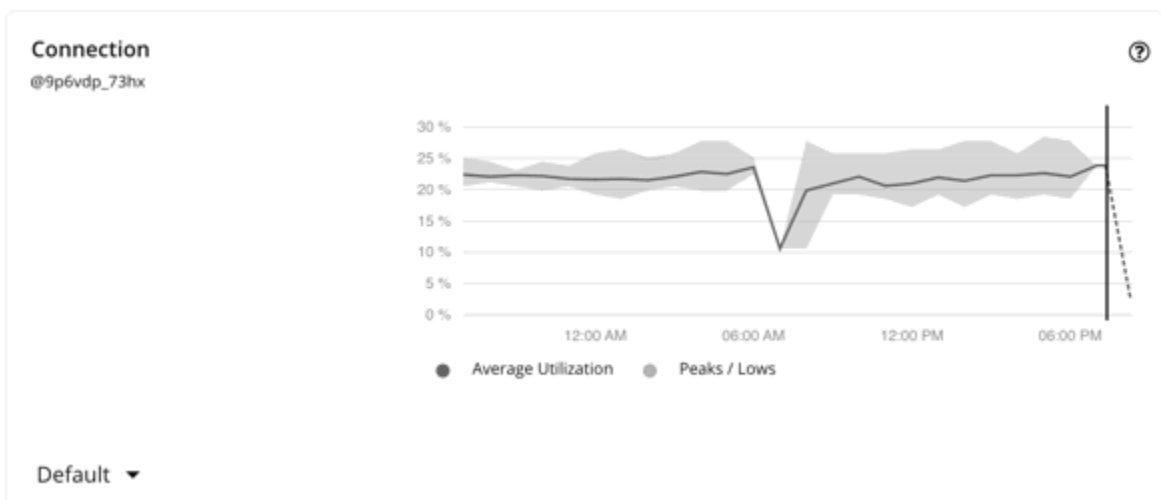
For example, if energy consumption for a host is 50 Wh, carbon footprint will be calculated as follows:

$$(50) * [(1.5) * (.25)] = 18.75 \text{ grams of CO2e}$$

## Connection Chart

Connection is the measurement of database connections utilized by applications.

Turbonomic collects connection data from Database Servers discovered via Databases, APM, and Cloud targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the Connection chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Database Server (see the next section for a list of supported Database Servers).

**Supported Database Servers**

Data is available in the Connection chart for Database Servers discovered via the following targets:

| Target      | Supported Database Servers                      |
|-------------|-------------------------------------------------|
| AWS         | RDS                                             |
| Azure       | SQL                                             |
| AppDynamics | MongoDB                                         |
| Instana     | Oracle                                          |
| MySQL       | MySQL                                           |
| Oracle      | Oracle                                          |
| JBoss       | All Database Servers discovered from the target |
| SQL         | SQL                                             |
| Tomcat      | All Database Servers discovered from the target |
| WebLogic    | All Database Servers discovered from the target |
| WebSphere   | All Database Servers discovered from the target |

**Effect on Memory Resize/Scale Actions**

Turbonomic uses connection data to generate memory resize actions for on-prem Database Servers.

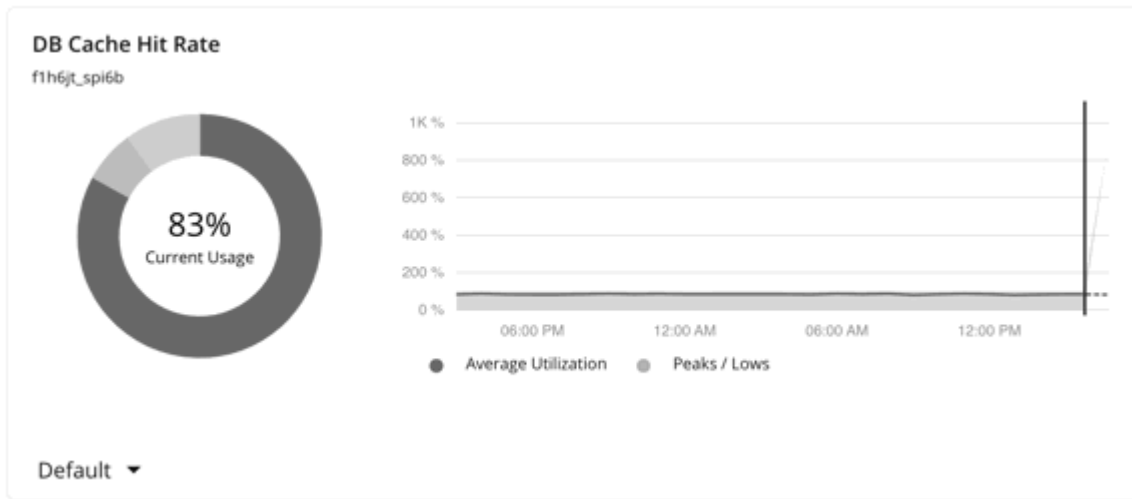
For cloud Database Servers, Turbonomic uses connection data as a constraint when generating scale actions. For details about scale actions, see [Database Server Actions](#).

*DB Cache Hit Rate Chart*

DB cache hit rate is the measurement of Database Server accesses that result in cache hits, measured as a percentage of hits versus total attempts. A high cache hit rate indicates efficiency.

Turbonomic collects cache hit rate data from Database Servers discovered via Databases, APM, and Cloud targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the DB Cache Hit Rate chart.





The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Database Server (see the next section for a list of supported Database Servers).

**Supported Database Servers**

Data is available in the DB Cache Hit Rate chart for Database Servers discovered via the following targets:

| Target      | Supported Database Servers |
|-------------|----------------------------|
| AWS         | RDS                        |
| Azure       | SQL                        |
| AppDynamics | SQL, Oracle                |
| Dynatrace   | SQL                        |
| Instana     | MySQL, SQL, Oracle         |
| MySQL       | MySQL                      |
| New Relic   | SQL, MySQL                 |
| Oracle      | Oracle                     |
| SQL         | SQL                        |

**Effect on Memory Resize Actions**

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.

A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal

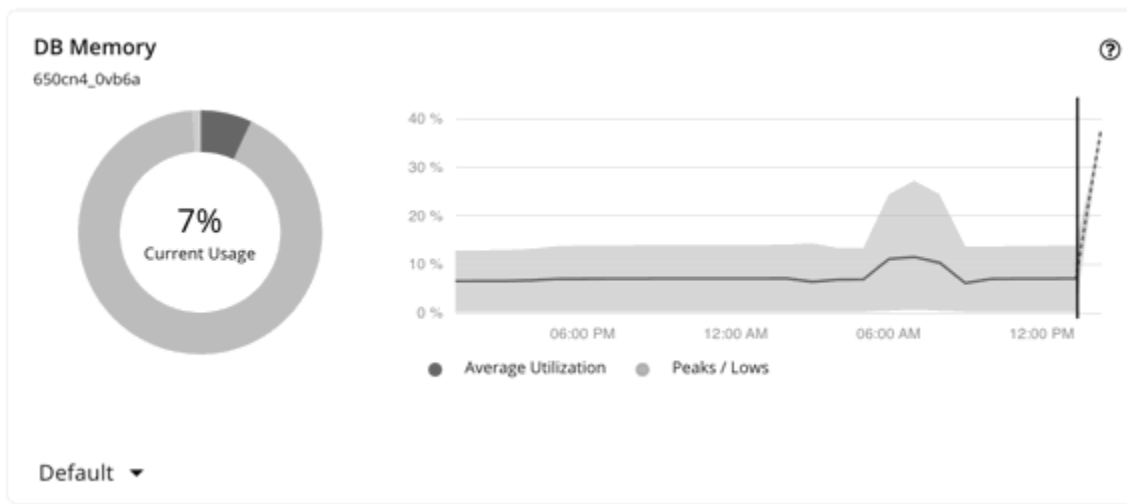
value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.

When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.

### DB Memory Chart

Database memory (or DBMem) is the measurement of memory utilized by a Database Server.

Turbonomic collects memory data from Database Servers discovered via Databases and APM targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the DB Memory chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

### Note:

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Database Server (see the next section for a list of supported Database Servers).

## Supported Database Servers

Data is available in the DB Memory chart for Database Servers discovered via the following targets:

| Target      | Supported Database Servers |
|-------------|----------------------------|
| AppDynamics | Oracle, MongoDB            |
| Dynatrace   | SQL, MySQL                 |
| Instana     | MySQL, SQL                 |
| MySQL       | MySQL                      |
| Oracle      | Oracle                     |
| SQL         | SQL                        |

## Memory Resize Actions

Actions to resize database memory are driven by data on the Database Server, which is more accurate than data on the hosting VM. Turbonomic uses database memory and cache hit rate data to decide whether resize actions are necessary.

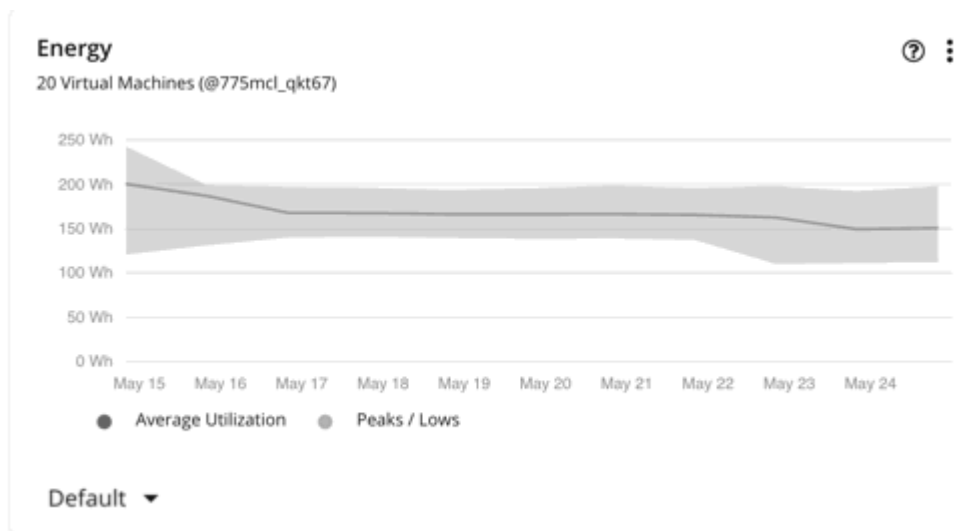
A high cache hit rate value indicates efficiency. The optimal value is 100% for on-prem (self-hosted) Database Servers, and 90% for cloud Database Servers. When the cache hit rate reaches the optimal value, no action generates even if database memory utilization is high. If utilization is low, a resize down action generates.

When the cache hit rate is below the optimal value but database memory utilization remains low, no action generates. If utilization is high, a resize up action generates.

### Energy Chart

Energy is the measurement of electricity consumed by a given entity, expressed in watt-hours (Wh).

Turbonomic collects energy-related data from hosts and VMs at 10-minute intervals. For VMs, the data collected from hosts is used to calculate VM energy consumption. Currently, Turbonomic collects data via vCenter, UCS Manager, and HPE OneView targets. When you set the scope to one or several hosts or VM discovered from these targets, the data that Turbonomic collected or calculated displays in the Energy chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

### Note:

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

Turbonomic also relies on host power data for the supported targets. This data is mandatory for Energy and Carbon Footprint calculations.

## Supported Entities

Data is available in the Energy chart for entities discovered via the following targets:

| Target Type | Target      | Supported Entities    |
|-------------|-------------|-----------------------|
| Fabric      | HPE OneView | Host, Virtual Machine |
|             | UCS Manager | Host                  |

| Target Type | Target  | Supported Entities    |
|-------------|---------|-----------------------|
| Hypervisor  | vCenter | Host, Virtual Machine |

## Calculation of Energy Consumed by VMs

Turbonomic calculates energy consumption for each VM on a host, based on two types of host energy:

- Host idle energy – this is the energy consumed by a host while in an idle state, when no processes are actively executing. This includes host overhead energy, which represents energy that is consumed by non-VM resources, such as hypervisors.

Host idle energy is attributed to all member VMs based on each VM's allocated capacity.

- Host active energy – this is the energy consumed by the VMs' operating systems and active processes.

Host active energy is attributed to all member VMs based on each VM's usage of host CPU.

The calculation can be expressed as follows:

$$\begin{aligned} & (\text{VM Size} / \text{Total Size for all VMs}) * \text{Host Idle Energy for VMs} + \\ & (\text{VM CPU Utilization} / \text{Total CPU Utilization for all VMs}) * \text{Host Active Energy for VMs} \\ & = \text{VM Energy Consumption (in Wh)} \end{aligned}$$

### Note:

Turbonomic rounds the calculated values.

For example, consider a host with three VMs. Currently, host *idle* energy for VMs is 50 Wh, while host *active* energy for VMs is 20 Wh.

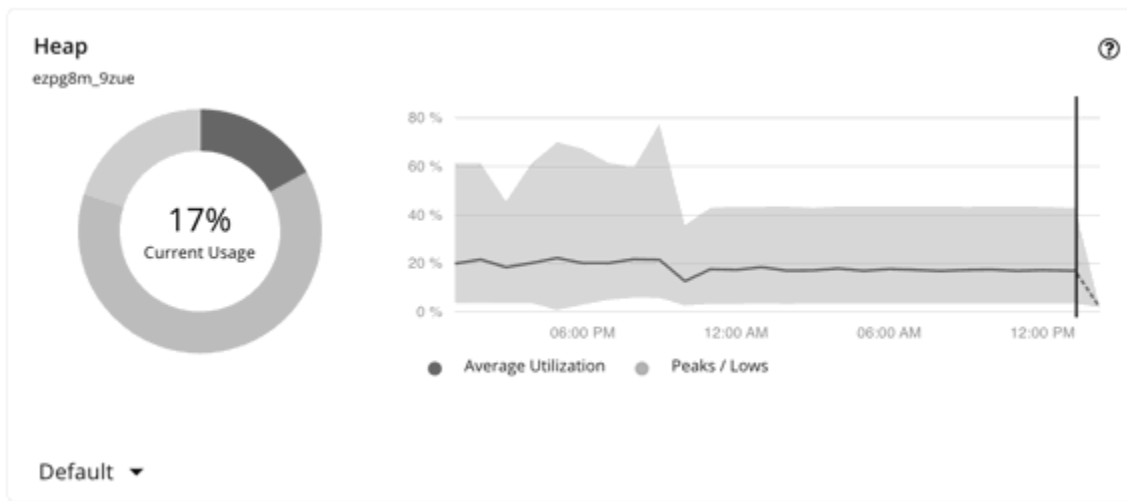
Turbonomic calculates VM energy consumption as follows:

| VM           | Size (Cores) | CPU Utilization (%)  | VM Energy Consumption (Wh)                        |
|--------------|--------------|----------------------|---------------------------------------------------|
| VM_01        | 2            | 50 (x 2 cores) = 100 | $(2 / 4) * 50 + (100 / 180) * 20 = \mathbf{36.1}$ |
| VM_02        | 1            | 80 (x 1 core) = 80   | $(1 / 4) * 50 + (80 / 180) * 20 = \mathbf{21.4}$  |
| VM_03        | 1            | 0 (idle)             | $(1 / 4) * 50 + (0) * 20 = \mathbf{12.5}$         |
| <b>Total</b> | <b>4</b>     | <b>180</b>           | <b>70</b>                                         |

### Heap Chart

Heap is the portion of a VM or container's memory allocated to individual applications.

Turbonomic collects heap data from Application Components discovered via Applications and APM targets. When you set the scope to one or several Application Components, the data that Turbonomic collected displays in the Heap chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Application Component (see the next section for a list of supported Application Components).

**Supported Application Components**

Data is available in the Heap chart for Application Components discovered via the following targets:

| Target      | Supported Application Components |
|-------------|----------------------------------|
| AppDynamics | Java applications, .NET, Node.js |
| Dynatrace   | Java applications                |
| Instana     | Java applications                |
| JBoss       | Java applications                |
| JVM         | Java applications                |
| New Relic   | Java applications, Node.js       |
| Tomcat      | Java applications                |
| WebLogic    | Java applications                |
| WebSphere   | Java applications                |

**Heap Resize Actions**

Turbonomic generates Heap resize actions if an Application Component provides Heap and Remaining GC Capacity, and the underlying VM or container provides VMem. These actions are recommend-only and can only be executed outside Turbonomic.

**Note:**

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

### Number of Replicas Chart

This chart shows the replicas of Application Components running over a given time period.

Use this chart if:

- SLO-driven scaling is enabled for a Service, and *provision* or *suspend* actions are executed by Turbonomic. These actions adjust the number of replicas to help you meet your SLO goals.

Or

- Kubernetes Horizontal Pod Autoscaler (HPA) is enabled for a *Deployment*, *ReplicaSet*, or *StatefulSet* that is exposed as a Service. Turbonomic discovers adjustments to the number of replicas made by HPA.

The chart shows following information:

- **Capacity** values

The number of desired pod replicas configured in the Workload Controller that backs the Service. This can be configured in *Deployment*, *ReplicaSet*, *StatefulSet*, *ReplicationController* or *DeploymentConfig*.

The chart plots the *maximum* observed capacity within the given time period.

- **Used** values

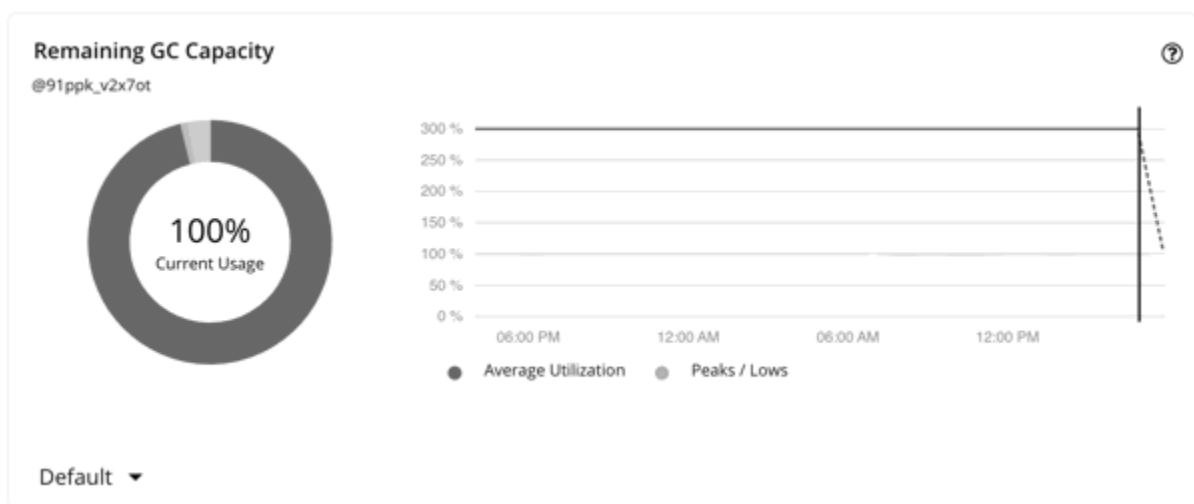
The number of *ready* pods owned by the Workload Controller. Pods in other states (for example, pending pods) are not counted.

The chart plots the *average* used values within the given time period. Hover over the chart to see the minimum and maximum used values.

### Remaining GC Capacity Chart

Remaining GC capacity is the measurement of Application Component uptime that is *not* spent on garbage collection (GC).

Turbonomic collects GC data from Application Components discovered via Applications and APM targets, and then uses that data to calculate remaining GC capacity. When you set the scope to one or several Application Components, the capacity that Turbonomic calculated displays in the Remaining GC Capacity chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Application Component (see the next section for a list of supported Application Components).

## Supported Application Components

Data is available in the Remaining GC Capacity chart for Application Components discovered via the following targets:

| Target      | Supported Application Components |
|-------------|----------------------------------|
| AppDynamics | Java applications, .NET          |
| Dynatrace   | Java applications                |
| Instana     | Java applications                |
| JBoss       | Java applications                |
| JVM         | Java applications                |
| New Relic   | Java applications, Node.js       |
| Tomcat      | Java applications                |
| WebLogic    | Java applications                |
| WebSphere   | Java applications                |

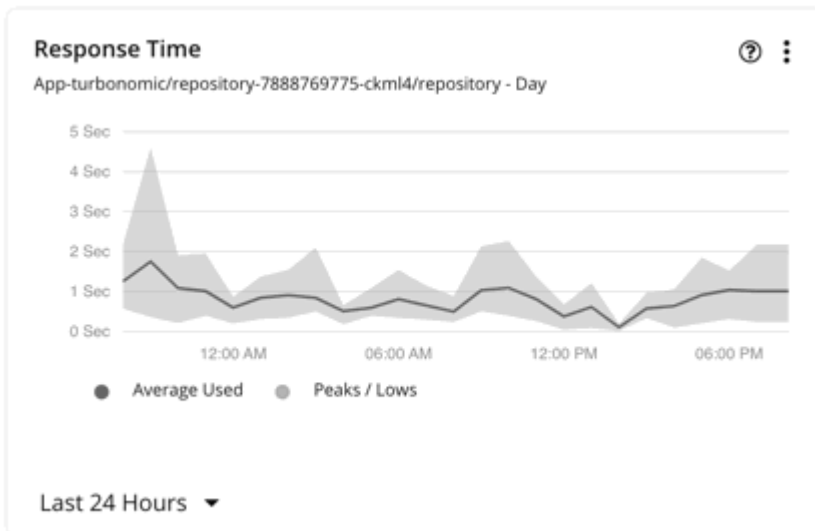
## Effect on Heap Resize Actions

Turbonomic generates Heap resize actions if an Application Component provides Heap and Remaining GC Capacity, and the underlying VM or container provides VMem. These actions are recommend-only and can only be executed outside Turbonomic.

### *Response Time Chart*

Response Time is the elapsed time between a request and the response to that request. Response Time is typically measured in seconds (s) or milliseconds (ms).

Turbonomic collects response time data from entities discovered via Applications, Databases, and APM targets. Entities include Business Applications, Business Transactions, Services, Application Components, and self-hosted Database Servers. When you set the scope to any of these entities, the data that Turbonomic collected displays in the Response Time chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

**Supported Entities**

Data is available in the Response Time chart for the following entities:

| Target      | Supported Entities                                                                          |
|-------------|---------------------------------------------------------------------------------------------|
| AppDynamics | Business Application, Business Transaction, Service, Application Component                  |
| Datadog     | Business Application, Business Transaction, Service, Application Component                  |
| Dynatrace   | Business Application, Service                                                               |
| Instana     | Business Application, Business Transaction, Service, Application Component, Database Server |
| JBoss       | Application Component                                                                       |
| JVM         | Application Component                                                                       |
| MySQL       | Database Server                                                                             |
| New Relic   | Business Transaction, Service, Application Component, Database Server                       |
| Oracle      | Database Server                                                                             |
| SQL         | Database Server                                                                             |
| Tomcat      | Application Component                                                                       |



| Target    | Supported Entities    |
|-----------|-----------------------|
| WebLogic  | Application Component |
| WebSphere | Application Component |

## Response Time SLOs

To evaluate the performance of your applications and Database Servers, set Response Time SLOs (Service Level Objectives) as an operational constraint in policies. For applications, you can set the SLO at the Business Application, Business Transaction, Service, or Application Component level.

**OPERATIONAL CONSTRAINTS**

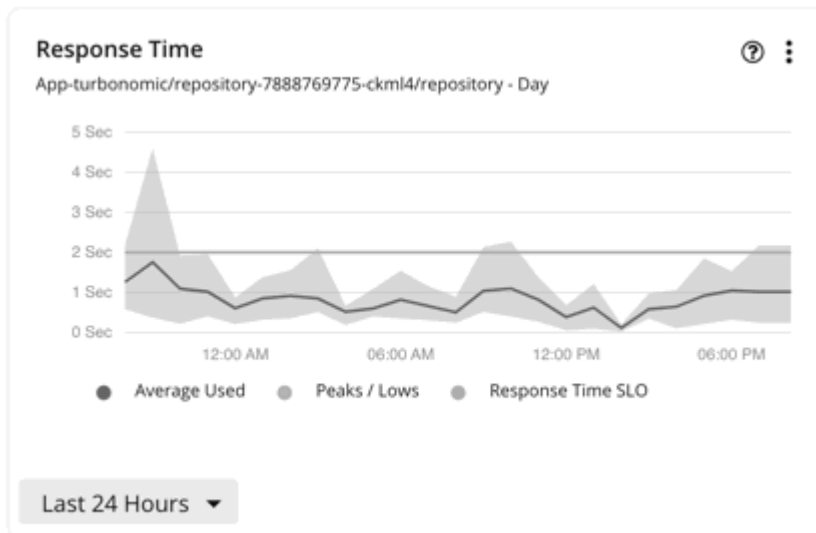
Transaction SLO

Enable Transaction SLO

Enable Response Time SLO

Response Time SLO [ms]  ms

After you create a policy, the SLO value appears as a solid straight line in the Response Time chart. You can then gauge performance against the given SLO.



If you do not set an SLO, Turbonomic estimates SLO based on historical Response Time data collected from the target, and then displays the estimated value in the Capacity and Usage chart, as Response Time capacity. This estimated value is *not* reflected in the Response Time chart.

| Capacity and Usage                  |            |            |             |
|-------------------------------------|------------|------------|-------------|
| SQLServer [win-dynatrace-mssql2017] |            |            |             |
| Commodity                           | Capacity   | Used       | Utilization |
| DB Cache Hit Rate                   | 100 %      | 100 %      | 100%        |
| Response Time                       | 44.76 msec | 42.65 msec | 95.3%       |
| TransactionLog                      | 347.21 MB  | 71.36 MB   | 20.55%      |
| Transaction                         | 3.58 TPS   | 0.38 TPS   | 10.66%      |
| DB Memory                           | 6.18 GB    | 644.22 MB  | 10.19%      |

SHOW ALL >

**Note:**

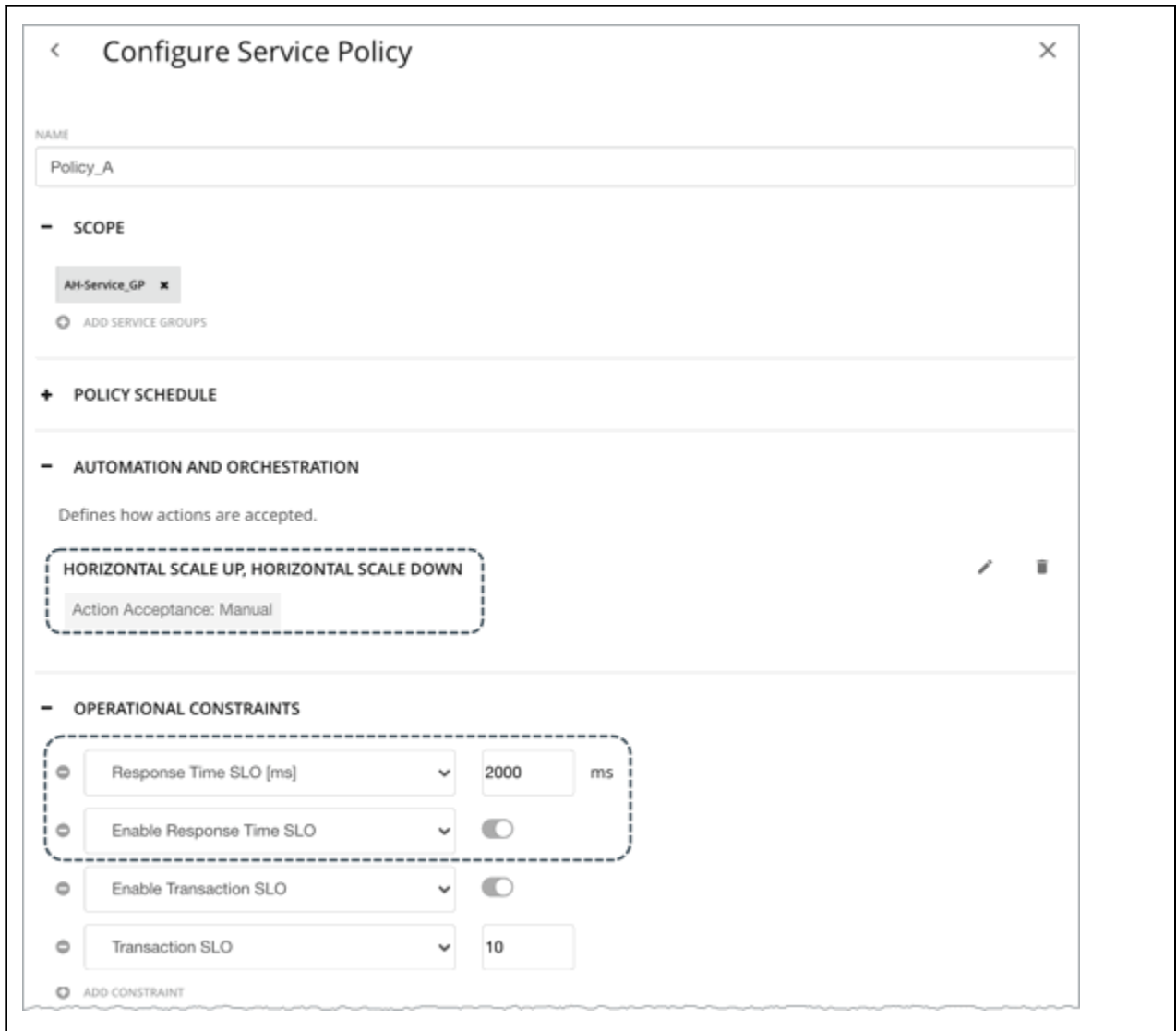
When you set an SLO value, Response Time capacity in the Capacity and Usage chart shows as N/A.

**Response Time SLOs for Kubernetes Services**

When you add a Kubernetes target, Turbonomic discovers container platform entities, including Kubernetes Services monitored by AppDynamics, Instana, Dynatrace, and New Relic.

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

To generate actions that adjust pod replicas, Kubernetes Services must be discovered by the Kubeturbo pod that you have deployed in your environment, as well as collect performance metrics via Instana or DIF (Data Ingestion Framework). In addition, Turbonomic requires that you turn on horizontal scaling and specify Response Time SLOs in policies for the affected Services.



Response Time SLO is the desired *weighted average* response time (in milliseconds) of all Application Component replicas associated with a Service.

**Note:**

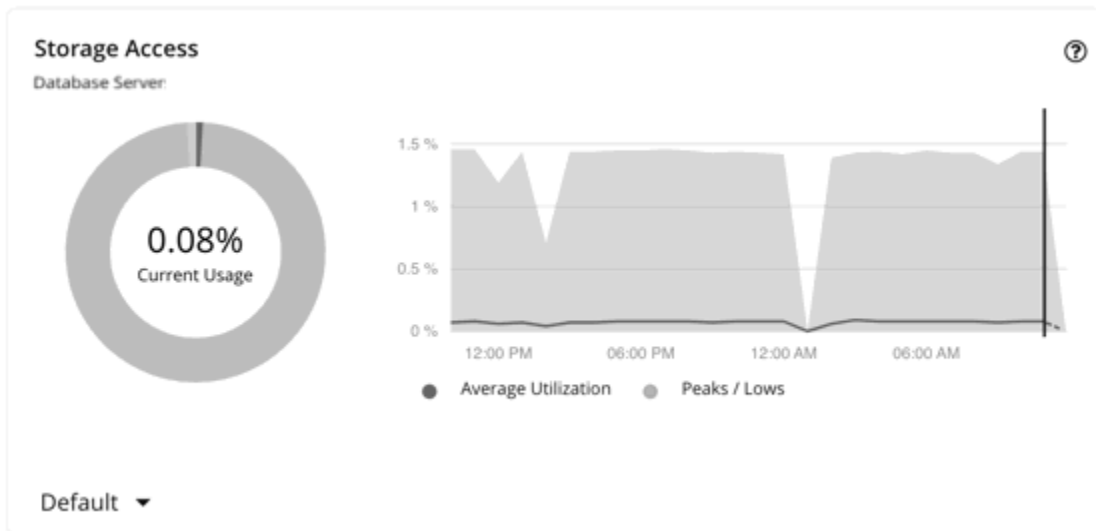
If you specified SLOs but turned off horizontal scaling in policies, no actions generate but SLO values will continue to display in the Response Time chart for Services, for your reference. This allows you to gauge performance against those SLOs.

For additional information, see [Actions for Kubernetes Services](#).

*Storage Access Chart*

Storage Access, also known as IOPS, is the per-second measurement of read and write access operations on a storage entity.

Turbonomic collects storage access data from VMs, Database Servers, and storage entities discovered via cloud, on-prem, and storage targets. When you set the scope to these entities, the data that Turbonomic collected displays in the Storage Access chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

**Supported Entities**

Data is available in the Storage Access chart for entities discovered via the following targets:

| Target Type    | Target      | Supported Entities                       |
|----------------|-------------|------------------------------------------|
| Cloud          | AWS         | Virtual Machine, Database Server, Volume |
|                | Azure       | Virtual Machine, Database Server, Volume |
|                | GCP         | Virtual Machine                          |
| Fabric         | HPE OneView | Virtual Machine, Storage                 |
| Hyperconverged | HyperFlex   | Storage, Disk Array                      |
|                | Nutanix     | Storage, Disk Array                      |
| Hypervisor     | Hyper-V     | Virtual Machine, Storage                 |
|                | vCenter     | Virtual Machine, Storage                 |

| Target Type | Target          | Supported Entities                |
|-------------|-----------------|-----------------------------------|
| Storage     | EMC ScaleIO     | Storage, Disk Array               |
|             | EMC VMAX        | Storage, Disk Array, Logical Pool |
|             | EMC XtremIO     | Storage, Disk Array               |
|             | HPE 3PAR        | Storage, Disk Array, Logical Pool |
|             | IBM FlashSystem | Storage, Disk Array, Logical Pool |
|             | NetApp          | Storage, Disk Array               |
|             | Pure Storage    | Storage, Disk Array               |

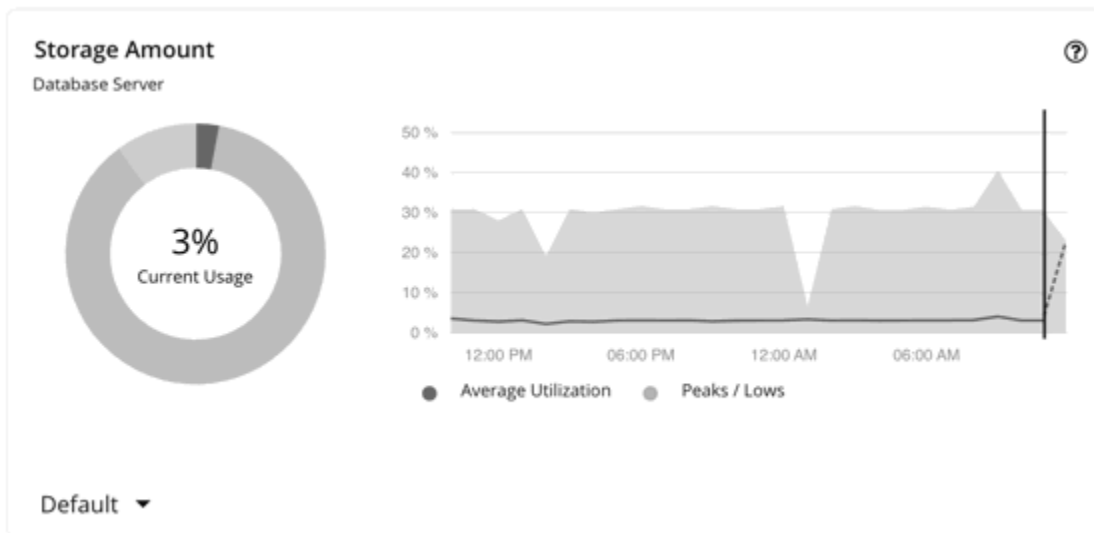
## Scale Actions

Turbonomic considers storage access (IOPS) utilization when recommending scaling actions for [cloud VMs](#), [cloud Database Servers](#), and [volumes](#).

### Storage Amount Chart

Storage Amount is the measurement of storage capacity utilized by the entity.

Turbonomic collects storage amount data from VMs, Database Servers, and storage entities discovered via cloud, on-prem, and storage targets. When you set the scope to these entities, the data that Turbonomic collected displays in the Storage Amount chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

### Note:

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

## Supported Entities

Data is available in the Storage Amount chart for entities discovered via the following targets:

| Target Type    | Target          | Supported Entities                                    |
|----------------|-----------------|-------------------------------------------------------|
| Cloud          | AWS             | Virtual Machine, Database Server                      |
|                | Azure           | Virtual Machine                                       |
|                | GCP             | Virtual Machine                                       |
| Fabric         | HPE OneView     | Storage                                               |
| Hyperconverged | HyperFlex       | Storage, Disk Array                                   |
|                | Nutanix         | Storage, Disk Array, Storage Controller               |
| Hypervisor     | Hyper-V         | Storage                                               |
|                | vCenter         | Storage                                               |
| Storage        | EMC ScaleIO     | Storage, Disk Array, Storage Controller               |
|                | EMC VMAX        | Storage, Disk Array, Logical Pool, Storage Controller |
|                | EMC XtremIO     | Storage, Disk Array, Storage Controller               |
|                | HPE 3PAR        | Storage, Disk Array, Logical Pool, Storage Controller |
|                | IBM FlashSystem | Storage, Disk Array, Logical Pool, Storage Controller |
|                | NetApp          | Storage, Disk Array, Storage Controller               |
|                | Pure Storage    | Storage, Disk Array, Storage Controller               |

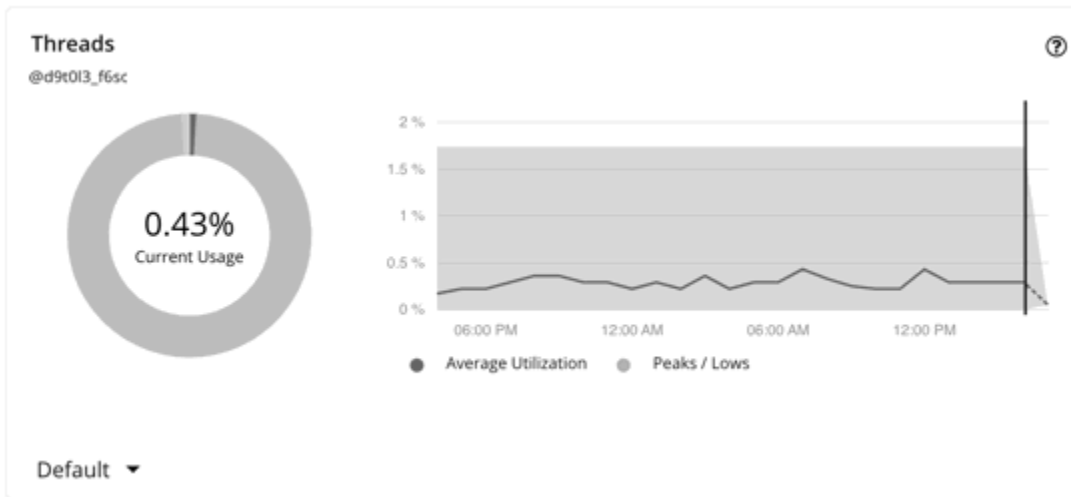
## Scale Actions

Turbonomic can recommend scaling [cloud VMs](#) and [cloud Database Servers](#) to instance types that can adequately meet storage demand at the lowest possible cost. For cloud Database Servers, Turbonomic can also recommend scaling up storage amount within the same instance type. Note that scaling up storage amount is non-disruptive but irreversible.

### *Threads Chart*

Threads is the measurement of thread capacity utilized by applications.

Turbonomic collects thread data from Application Components discovered via Applications and APM targets. When you set the scope to one or several Application Components, the data that Turbonomic collected displays in the Threads chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Application Component (see the next section for a list of supported Application Components).

**Supported Application Components**

Data is available in the Threads chart for Application Components discovered via the following targets:

| Target      | Supported Application Components |
|-------------|----------------------------------|
| AppDynamics | Java applications, .NET          |
| JBoss       | Java applications                |
| JVM         | Java applications                |
| New Relic   | Java applications                |
| Tomcat      | Java applications                |
| WebLogic    | Java applications                |
| WebSphere   | Java applications                |

**Thread Pool Resize Actions**

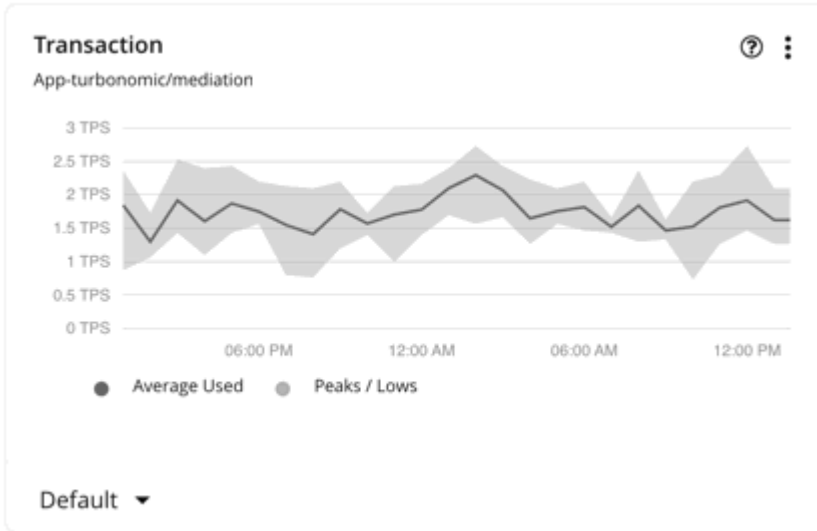
Turbonomic generates thread pool resize actions. These actions are recommend-only and can only be executed outside Turbonomic.

*Transaction Chart*

Transaction is a value that represents the per-second utilization of the transactions allocated to a given entity.

Turbonomic collects transaction data from entities discovered via Applications, Databases, and APM targets. Entities include Business Applications, Business Transactions, Services, Application Components,

and self-hosted Database Servers. When you set the scope to any of these entities, the data that Turbonomic collected displays in the Transaction chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

**Supported Entities**

Data is available in the Transaction chart for the following entities:

| Target      | Supported Entities                                                                          |
|-------------|---------------------------------------------------------------------------------------------|
| AppDynamics | Business Application, Business Transaction, Service, Application Component, Database Server |
| Dynatrace   | Business Application, Service, Database Server                                              |
| Instana     | Business Application, Business Transaction, Service, Database Server                        |
| JBoss       | Application Component                                                                       |
| MySQL       | Database Server                                                                             |
| New Relic   | Business Transaction, Service, Application Component, Database Server                       |
| Oracle      | Database Server                                                                             |
| SQL         | Database Server                                                                             |
| Tomcat      | Application Component                                                                       |
| WebLogic    | Application Component                                                                       |



| Target    | Supported Entities    |
|-----------|-----------------------|
| WebSphere | Application Component |

## Transaction SLOs

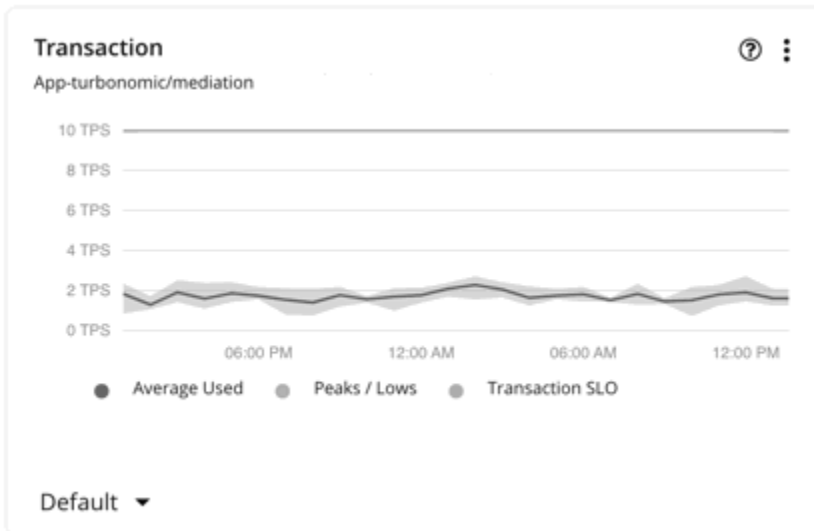
To evaluate the performance of your applications and Database Servers, set Transaction SLOs as an operational constraint in policies. For applications, you can set the SLO at the Business Application, Business Transaction, Service, or Application Component level.

**OPERATIONAL CONSTRAINTS**

Enable Response Time SLO   ms

Enable Transaction SLO

After you create a policy, the SLO value appears as a solid straight line in the Transaction chart. You can then gauge performance against the given SLO.



If you do not set an SLO, Turbonomic estimates SLO based on historical Transaction data collected from the target, and then displays the estimated value in the Capacity and Usage chart, as Transaction capacity. This estimated value is *not* reflected in the Transaction chart.

| Capacity and Usage       |          |            |             |
|--------------------------|----------|------------|-------------|
| App-turbonomic/mediation |          |            |             |
| Commodity                | Capacity | Used       | Utilization |
| Remaining GC Capacity    | 100 %    | 99.59 %    | 99.59%      |
| Transaction              | 3.2 TPS  | 2.2 TPS    | 68.79%      |
| Heap                     | 24 GB    | 0.93 GB    | 3.87%       |
| Virtual Memory           | 32 GB    | 1.54 GB    | 4.82%       |
| Virtual CPU              | 17.6 GHz | 835.07 MHz | 4.74%       |

SHOW ALL >

**Note:**

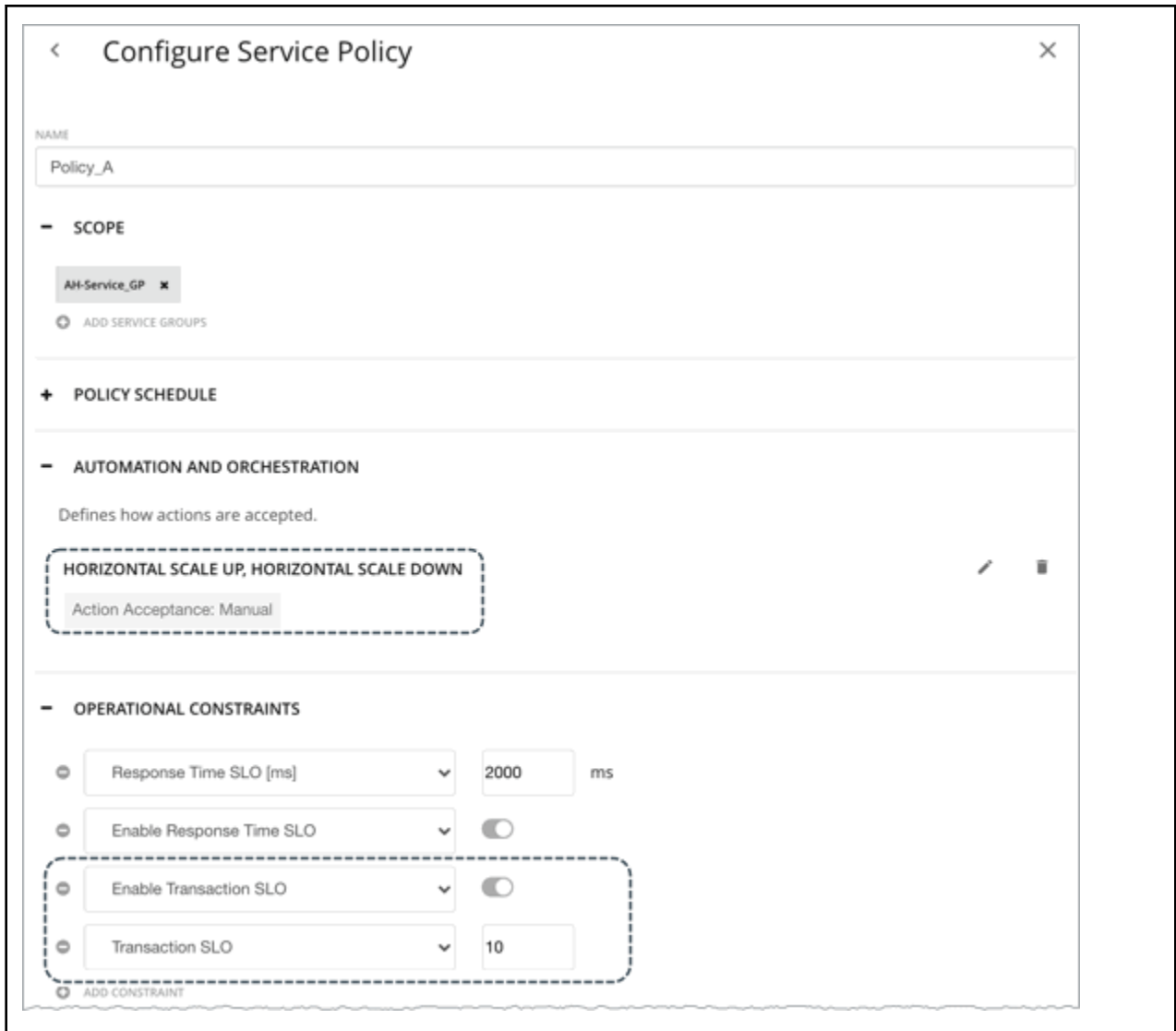
When you set an SLO value, Transaction capacity in the Capacity and Usage chart shows as N/A.

**Transaction SLOs for Kubernetes Services**

When you add a Kubernetes target, Turbonomic discovers container platform entities, including Kubernetes Services managed by AppDynamics, Instana, Dynatrace, and New Relic.

For horizontally scalable Kubernetes Services that collect performance metrics (or KPIs) for applications, Turbonomic can dynamically adjust the number of pod replicas that back those Services to help you meet SLOs (Service Level Objectives) for your applications.

To generate actions that adjust pod replicas, Kubernetes Services must be discovered by the Kubeturbo pod that you have deployed in your environment, as well as collect performance metrics via Instana or DIF (Data Ingestion Framework). In addition, Turbonomic requires that you turn on horizontal scaling and specify Transaction SLOs in policies for the affected Services.



Transaction SLO is the maximum number of transactions per second that each Application Component replica can handle.

**Note:**

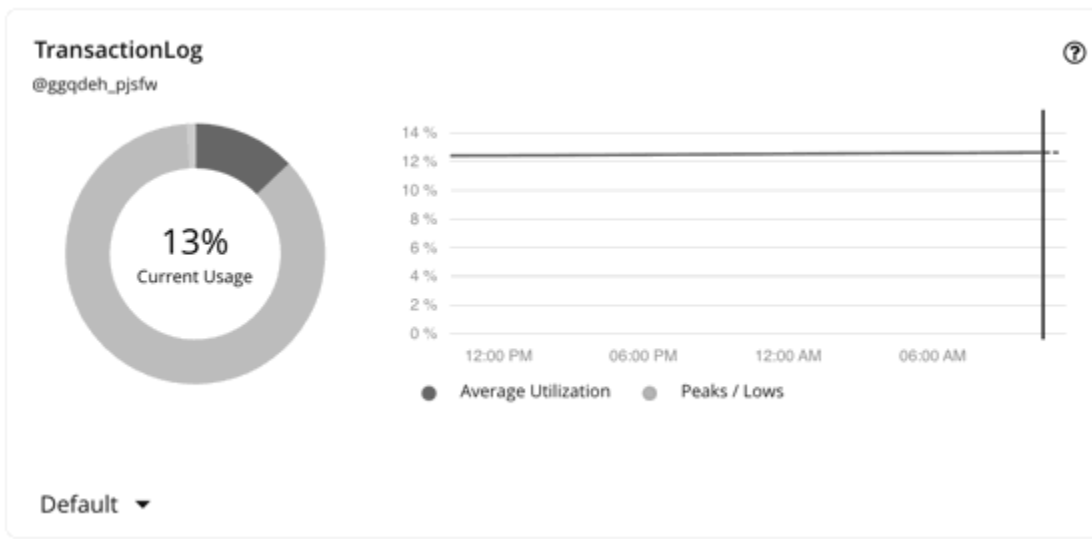
If you specified SLOs but turned off horizontal scaling in policies, no actions generate but SLO values will continue to display in the Transaction chart for Services, for your reference. This allows you to gauge performance against those SLOs.

For additional information, see [Actions for Kubernetes Services](#).

*Transaction Log Chart*

Transaction log is the measurement of storage capacity utilized by Database Servers for transaction logging.

Turbonomic collects transaction log data from Database Servers discovered via Databases and APM targets. When you set the scope to one or several Database Servers, the data that Turbonomic collected displays in the Transaction Log chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported Database Server (see the next section for a list of supported Database Servers).

**Supported Database Servers**

Data is available in the Transaction Log chart for Database Servers discovered via the following targets:

| Target      | Supported Database Server |
|-------------|---------------------------|
| AppDynamics | SQL                       |
| Oracle      | Oracle                    |
| SQL         | SQL                       |

**Transaction Log Resize Actions**

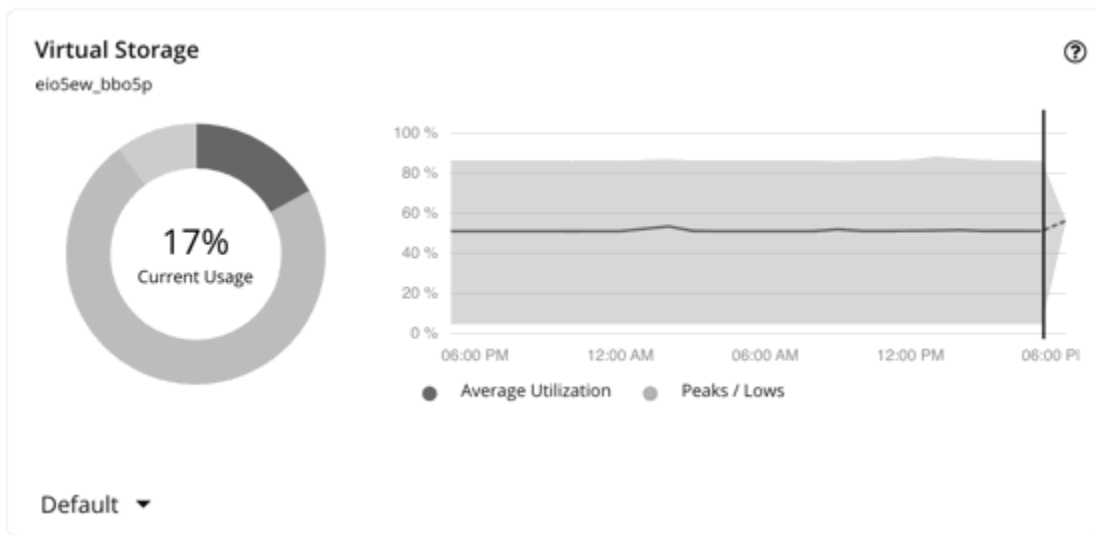
Resize actions based on the transaction log resource depend on support for virtual storage in the underlying hypervisor technology.

Currently, Turbonomic does not support resize actions for Oracle and Database Servers on the Hyper-V platform (due to the lack of API support for virtual storage).

*Virtual Storage Chart*

Virtual storage is the measurement of virtual storage capacity utilized by VMs.

Turbonomic collects virtual storage data from VMs discovered via on-prem and APM targets. When you set the scope to one or several VMs, the data that Turbonomic collected displays in the Virtual Storage chart.



The chart shows average and peak/low values over time. Use the selector at the bottom left section of the chart to change the time frame.

**Note:**

An empty chart could be the result of delayed discovery, target validation failure, unavailable data for the given time frame, or an unsupported entity (see the next section for a list of supported entities).

**Supported Entities**

Data is available in the Virtual Storage chart for entities discovered via the following targets:

| Target Type      | Target      | Supported Entities |
|------------------|-------------|--------------------|
| Database         | MySQL       | Virtual Machine    |
|                  | Oracle      | Virtual Machine    |
|                  | SQL         | Virtual Machine    |
| Fabric           | HPE OneView | Virtual Machine    |
| Guest OS Process | SNMP        | Virtual Machine    |
|                  | WMI         | Virtual Machine    |
| Hyperconverged   | HyperFlex   | Virtual Machine    |
|                  | Nutanix     | Virtual Machine    |
| Hypervisor       | Hyper-V     | Virtual Machine    |
|                  | vCenter     | Virtual Machine    |

**Virtual Storage Actions**

Turbonomic can recommend the following virtual storage actions:

- Move a VM's volume (virtual storage) due to excess utilization of the current datastore, or for more efficient utilization of datastores in the environment.

- Reconfigure a VM's volume (virtual storage) to comply with placement policies.

## ***Top Utilized Charts***

Top Utilized charts show the entities or groups with the most utilization.

## **Entity Type**

Entity types you can choose include:

- Accounts (public cloud)
- Business Applications
- Business Transactions
- Services
- Application Components
- Zones
- Chassis
- Clusters (of hosts)
- Containers
- Container Pods
- Container Specs
- Namespaces
- Workload Controllers
- Data Centers
- Databases
- Database Servers
- Disk Arrays
- IO Modules
- Internet
- Logical Pool
- Networks
- Hosts
- Resource Groups
- Regions
- Storage Devices
- Storage Controllers
- Switches
- Virtual Data Centers
- Virtual Machines
- Volumes
- Wasted Files

## **Data Type**

Depending on the entity type (for example, Clusters), you can choose **Headroom** or **Utilization** information in the chart.

## Commodity

Depending on the entity type, you can add one or more different resource commodities that you want to measure.

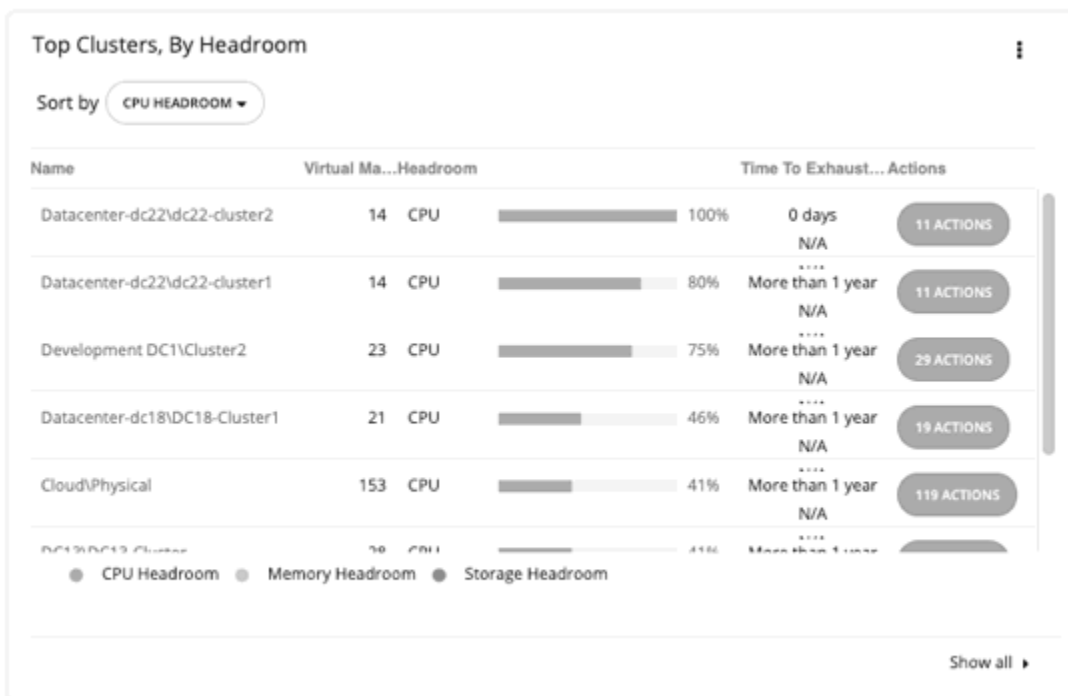
## Display

The chart lists the top entities by utilization of the commodities that you or the system has set. Depending on the entity type and scope, you can sort the information. To view the utilization details, hover over the entity to display the tooltip.

To drill down to an entity, click the entity name in the chart. This sets the scope to that entity.

Click the **ACTIONS** button for an entity to examine the actions that are pending for it, and then decide which ones are safe to execute.

Example: A top clusters chart which can be sorted by CPU headroom or CPU exhaustion.



### Top Clusters Chart

This chart shows the top clusters in your on-prem environment by CPU, memory, and storage capacity or utilization. In the default view, the chart shows the top clusters by CPU headroom (available capacity). It also shows time to exhaustion of cluster resources, which is useful for future planning (for example, you might need to buy more hardware).

To calculate cluster capacity and headroom, Turbonomic runs nightly plans that take into account the conditions in your current environment. The plans use the Economic Scheduling Engine to identify the optimal workload distribution for your clusters. This can include moving your current VMs to other hosts within the given cluster, if such moves would result in a more desirable workload distribution. The result of the plan is a calculation of how many more VMs the cluster can support.

Click the **ACTIONS** button for a given cluster to see the actions that Turbonomic recommends to keep cluster resources in the desired state, and then decide which ones are safe to execute.

Click **Show All** to see all of the clusters. In the Show All list, you can also download capacity data as a CSV file. Click a cluster name to set the scope to that cluster and view more details about its current capacity and health.

### *Top Accounts Chart*

This chart lists the cloud accounts with the largest potential savings. These are the savings you would realize if you execute pending actions for your cloud workloads. Click the **ACTIONS** button to examine these actions and decide which ones are safe to execute. The chart also shows billed costs in the last 30 days.

To set the scope to a particular account, click the account name.

Click **Show all** to view additional information, including the number of actions that have been executed for individual accounts or workloads, along with the resulting savings. If you have multiple cloud providers, each provider will have its own tab. You can download the accounts list as a CSV file.

## **AWS Accounts**

The chart shows the AWS master and member accounts that you have added as targets, including [GovCloud](#) accounts. Accounts with a star symbol are master accounts.

For GovCloud accounts, the billed cost is shown as **N/A** since invoicing for these accounts is managed through an associated AWS *standard account*. Adding this standard account as a target is optional. When added, it appears in the Top Accounts chart and shows the total billed cost for the GovCloud accounts and the standard account itself.

### **Note:**

Specific RIs can provide savings for multiple accounts. However, individual accounts show the full RI savings, which can result in exaggerated savings for that account.

## **Azure Accounts**

The chart shows the subscriptions discovered via the service principal accounts that you have added as targets, including [Azure Government](#) subscriptions.

## **GCP Accounts**

The chart shows the folders and projects discovered via the GCP service accounts that you have added as targets.

If a service account has access to a folder with projects and subfolders, the folder displays as the top-level account. Click **Show All** to see the full resource hierarchy and top-down data. If a service account has access to a project or subfolder but not its parent folder, the project or subfolder displays as the top-level account.

### *Top Resource Groups Chart*

This chart highlights the estimated monthly cost for the top resource groups in your cloud environment and the savings you would realize if you execute the pending actions. Click the **ACTIONS** button to examine these actions and decide which ones are safe to execute. Click a resource group to set the scope to that group.

The chart also counts actions that have been executed for individual groups, and then shows the resulting savings.

## **Workload Health Charts**

Workload Health charts show the health of workloads from the compliance, efficiency improvement, and performance assurance perspectives. These charts use current (real-time) data for the workloads chosen for the chart widget scope.

## **Chart Type**

You can set the display to:



- Text
- Ring Chart
- Horizontal Bar

## Breakdown

You can choose:

- **Workload by Compliance**

A virtual environment can include policies that limit availability of resources. It's possible that the environment configuration violates these defined policies. In such cases, Turbonomic identifies the violation and recommends actions that bring the entity back into compliance.

- **Workload by Efficiency Improvement**

Efficient utilization of resources is an important part of running in the desired state. Running efficiently maximizes your investment and reduces cost. When Turbonomic discovers underutilized workloads, it recommends actions to optimize operations and save money.

- **Workload by Performance Assurance**

Ultimately, the reason to manage workloads in your environment is to assure performance and meet QoS goals. When Turbonomic detects conditions that directly put QoS at risk, it recommends associated actions to assure performance. You can consider these critical conditions, and you should execute the recommended actions as soon as possible.

Workload Health charts indicate actions that you should consider to improve the health of workloads. To see a list of actions, click **Show Actions** at the bottom of the chart.

## Environment Charts

Environment charts provide an overview of your environment. They show the targets that you are managing and count the entities that Turbonomic has discovered through those targets. For example, you can display the cloud service providers, hypervisors, and the number of workloads.

## Environment Type

You can choose one of the following views:

- Hybrid (both on-prem and cloud)
- Cloud
- On-Prem

## Display

The chart shows the information as a Text chart type.

## Workload Improvements Charts

Workload Improvements charts track the health of workloads in your environment over time, and map the health to the number of actions Turbonomic has executed in that time period.

In the chart, you can see the significance and value of executed actions:

- Workloads Overall  
This is the total number of workloads over time.
- Workloads with Performance Risks  
These are the workloads that are not performing well.
- Inefficient Workloads

These are the workloads that are running on under-utilized hosts or are not being utilized.

- Workloads Out of Compliance

These are the workloads that are violating a placement policy. Workloads that are not in compliance might be running on a host or placed on storage, for example, that violate a placement policy.

- Executed actions

Actions that Turbonomic executed.

The vertical line shows when the last data point was polled in your environment.

## Environment Type

You can choose one of the following views:

- Hybrid (both on-prem and cloud)
- Cloud
- On-Prem

## Display

The chart shows the information as a Line chart.

## Cloud Chart Types

These chart widgets provide information on the status of your cloud environment.

For many cloud chart widgets that display costs and savings, Turbonomic uses the billing reports from your cloud service providers to build a picture of your overall costs. The data includes all costs that the service provider includes in the billing report. Turbonomic parses these reports into the formats that it uses for the cloud chart widgets.


### Note:

In order for Turbonomic to access AWS monthly reports, you must have created a Cost and Usage report in your AWS account and you must store it in an S3 bucket.

## Billing Breakdown Charts

### Important:

This chart has been removed from the product and can no longer be added to dashboards. Delete it now and start using the replacement chart, Workload Cost Breakdown.

To delete this chart, click the ellipsis icon (  ) at the top right section of the chart and then select **Delete**.

To add the Workload Cost Breakdown chart, click **Add Widget** at the top right section of any dashboard, find and select **Expenses**, go to the Commodity field, and then select **Workload Cost Breakdown**.

## Expenses Charts

Turbonomic uses billing information from your cloud provider to [track](#) compute, storage, and license costs for your workloads. Use this chart to manage your costs and see how they evolve over time.

## Commodity

You can choose:

- **Billed Cost by Service Provider**

See costs over time for each cloud service provider that you use in your cloud environment.

You can open more than one account from a single service provider. If you are running workloads on different service providers, then this chart shows the distribution of costs across them.

- **Top Billed Cost by Account**

The chart shows the cloud accounts with the largest costs. The chart's legend displays up to 20 actual accounts and, if needed, an additional item labeled 'Others' that represents all accounts that are not in the top 20. Hover on a data point to see costs for individual accounts.

- **Top Billed Cost by Service**

The chart shows the services with the largest costs. The chart's legend displays up to 20 actual services and, if needed, an additional item labeled 'Others' that represents all services that are not in the top 20. Hover on a data point to see costs for individual services.

- **Workload Cost Breakdown**

This chart breaks down costs by the following categories:

- On-Demand Compute
- Spot Compute
- On-Demand Compute License Bundle – only applies to Azure VMs and AWS VMs running Windows OS
- On-demand License
- Reserved License
- Storage
- Other cost – includes workload costs that are not categorized by Turbonomic, such as network charges

Points to consider:

- For VMs fully covered by AWS Reserved Instances, Azure reservations, or GCP committed use discounts, on-demand cost is 0 (zero).
- For Azure VMs, the chart shows cost data from the day the VM was discovered by Turbonomic. Cost data before discovery is not available and therefore not reflected in the chart.

## **Chart Type**

You can set the display to:

- Line Chart
- Stacked Bar Chart
- Area Chart

## **Chart Time Frame**

Currently, the chart can display data from the last 7 or 30 days. As you change the time frame, Turbonomic divides the reported information into the appropriate time units to match that time frame. However, the source remains the same. Changing the time frame does not affect the source data or increase data polling.

## **Cloud Tier Breakdown Charts**

Cloud Tier charts show the cloud tiers that Turbonomic discovers for the chart widget scope. For example, if the Chart Widget Scope is set to All Cloud VMs and the Entity Type is set to Virtual Machine, the chart shows all the cloud tiers that the workloads use.

## **Entity Type**

You can choose any entity type in the list.

## **Chart Type**

You can set the display to:

- Text
- Ring Chart
- Horizontal Bar

### Location Charts

Location charts show cloud provider regions in a world map for which there are discovered workloads. Click on any region to examine more detailed information in a scoped view.

### Display

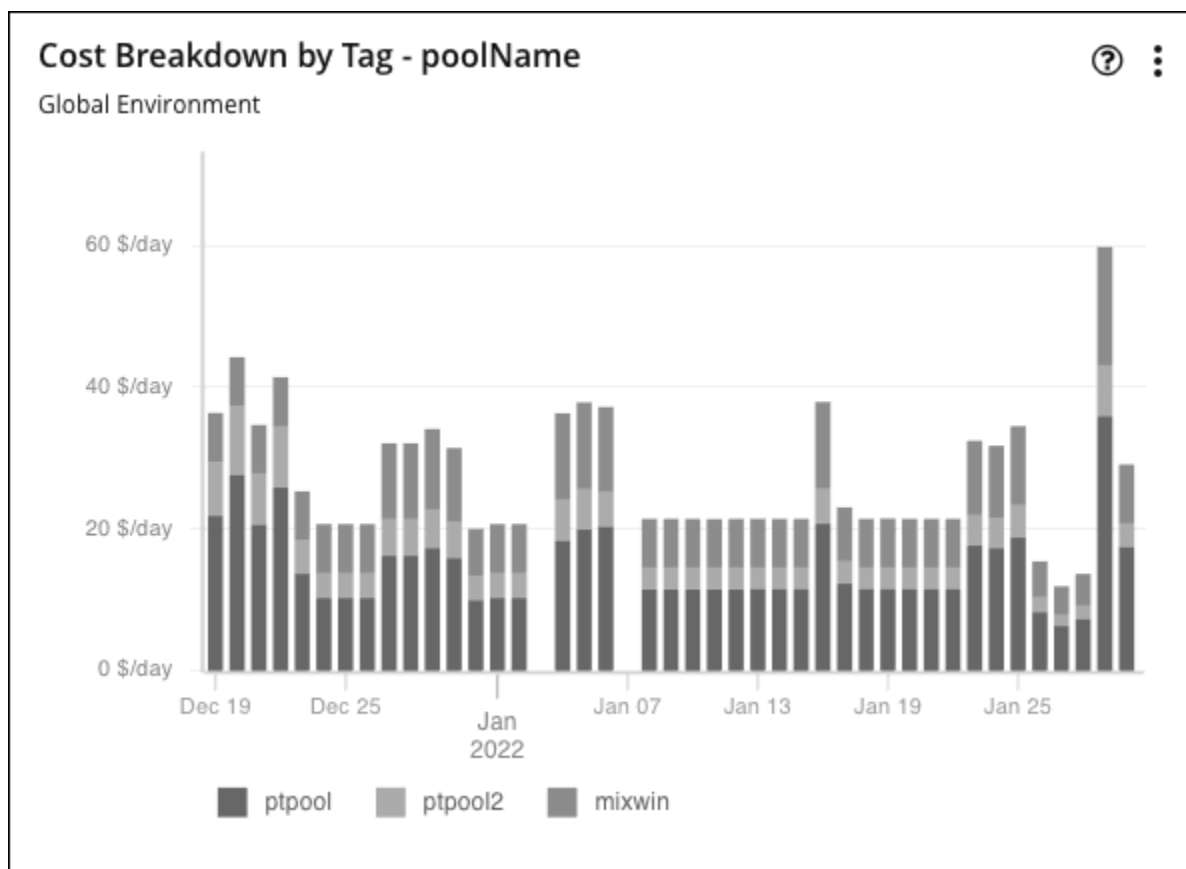
The chart shows the regions in countries in a Map chart.

### Cost Breakdown by Tag Charts

Cost Breakdown By Tag charts show the costs for tagged cloud entities that Turbonomic has discovered in your AWS, Azure, or GCP environment. For the tagged entities in scope, the chart shows how daily costs change over time.

You choose a tag key to track, and then choose which tag values to include in the chart. Each data point aggregates the costs for all the entities with a given tag/value pair. You can display the cost breakdown in a stacked bar chart or an area chart.

For example, in this stacked bar chart, the tag **poolName** is *workload-type* and the tag **Values** are *ptpool*, *ptpool2*, and *mixwin*.



## Timeframe

Currently, the chart can display data from the last 7 or 30 days.

## Chart Type

You can set the display to:

- Area Chart
- Stacked Bar Chart

For more detail, hover over a data point. A tooltip appears to show specific values for that date. Click the legend items to show/hide data for specific values.

## Tag Settings

Choose the Tag/Value pairs you want to display in the chart.

Note that tag Key and Value are case insensitive. Each data point in the chart aggregates the costs for all entities with the given tag Key/Value pair, regardless of case.

### • Key

The tag name that you want to chart. Turbonomic discovers the tags you have configured in your environment.

You can choose one Key for the chart.

### • Values

The values that you have configured in your environment for the given Key.

You can choose multiple values. To shorten the list of values, type a filter string in the Values field.

## ***Cumulative Savings and Investments Charts***

Actions for your cloud workloads usually have cost savings or investments attached to them. For example, deleting unattached volumes can lower your costs significantly (savings), while scaling a VM to a different tier to improve performance could incur additional costs (investments). These charts highlight total savings and investments as a result of executing actions. Turbonomic uses billing data from your cloud provider to calculate savings and investments.

These charts display in the built-in Cloud Executive Dashboard. You can add these charts to the default views in the Home Page or to your custom dashboards.

Points to consider:

- Billing data from your cloud provider generally takes 1 to 2 days to update fully. Savings and investment figures may change until the data is fully updated.
- The charts show savings and investments data for actions executed *after* the installation of version 8.9.3. Data for actions executed before this version will not be shown, but can be extracted from the platform using a script. If you need to extract the data, contact your Turbonomic representative for assistance.
- In the Cloud Executive Dashboard, scoped users can see data for the entire cloud environment.

## Scale Actions

Scale actions involve moving an entity to a different compute tier or adjusting the entity's allocated capacity. Currently, Turbonomic calculates savings and investments when scaling the following entities:

- VMs (AWS, Azure, and GCP)
- Volumes (AWS and Azure)
- Databases (Azure SQL vCore or DTU)

- Virtual Machine Specs (Azure App Service plans)

Turbonomic calculates savings and investment *per entity*. Calculation is based on the *before-action* and *after-action* costs.

- Before-action cost is the cost of an entity before an action was executed. This is based on the cost snapshot on the day of action execution.
- After-action cost is the cost of an entity as reflected in the daily billing report from the cloud provider.

Savings and investments are the total of all past scaling actions, with the exception that a scale in one direction (such as a scale up) reduces the amounts of previous actions in the opposite direction (such as a scale down), until one or more previous actions have no more effect.

To illustrate, consider three consecutive scale actions for a newly discovered VM and the effect of these actions on the calculated costs. Note that these actions are assumed to have been executed at midnight. Costs are calculated proportionately when actions are executed at other times of the day.

| Day | Executed Action | Before-action Cost | After-action Cost | Investments       | Savings           |
|-----|-----------------|--------------------|-------------------|-------------------|-------------------|
| 01  | Scale up        | \$2                | \$5               | $\$5 - \$2 = \$3$ | <b>\$0</b>        |
| 02  | Scale down      | \$5                | \$4               | $\$4 - \$2 = \$2$ | $\$5 - \$4 = \$1$ |
| 03  | Scale down      | \$4                | \$2               | $\$2 - \$2 = \$0$ | $\$5 - \$2 = \$3$ |

Investment is the difference between the after-action cost on a particular day and the lowest ever before-action cost up to that day (\$2 for all three days in the example). Savings is the difference between the largest ever before-action cost up to a particular day (\$5 on Days 02 and 03) and the after-action cost on that day.

**Note:**

When the result of the calculation is a negative amount, Turbonomic considers savings or investments to be \$0.

Turbonomic reads the daily billing report from the cloud provider to determine the current daily cost of an entity. Calculation also takes into account workload uptime (for VMs) and the effect of consecutive scale actions on the same workload.

Points to consider:

- Calculation can adjust to varying VM uptime and discount coverage over time.
- Calculation stops for terminated entities or entities that Turbonomic no longer discovers.
- Calculation stops if there is a cost change in the entity that is *not* the result of a Turbonomic action.


**Delete Actions**

Delete actions always result in savings. Currently, Turbonomic calculates savings when deleting the following entities:

- Unattached volumes (AWS, Azure, and GCP)
- Empty Virtual Machine Specs (Azure App Service plans)

Turbonomic checks the daily cost of an entity on the day the entity was deleted, and uses that cost to calculate savings from that day onwards. The daily bill from the cloud provider is not used to calculate savings.

**Chart Settings**

Click the More options icon (  ), and then select **Edit** to modify the following settings:

- Scope

By default, the charts are scoped to your global environment. You can change the scope to one or several accounts, billing families, groups, or entities.

- Timeframe

- Last 7 Days or Last 30 Days – Each data point in the chart shows total savings or investments from previous days until the given day.
- Last Year – Each data point in the chart shows total savings or investments from previous months until the given month.

- Chart Type

You can set the display to:

- Text and Bar Chart
- Text and Area Chart
- Stacked Bar Chart
- Area Chart
- Text

- Type

Switch between the **Cumulative Savings** and **Cumulative Investments** views. You can also change the displayed data to just **Savings** or **Investments** if you do not wish to see how the savings or investment costs accumulate over time.

- Group By

This setting breaks down data points by the selected group. By default, no group is selected. Each data point only shows the total cumulative values for a given day or month, depending on the selected timeframe.

The following groups are supported:

- Account – Cloud accounts discovered in your environment
- Action Type – Scale and delete actions
- Cloud Service Provider – AWS, Azure and GCP
- Entity Type – Virtual Machine, Volume, Database (Azure only), Virtual Machine Spec (Azure only)
- Region – Regions for the entities
- Resource Group (Azure only) – Available resource groups in your environment

For example, if you selected **Account**:

- The chart represents each account as a distinct color.
- When you hover on a data point, the chart shows a list of accounts, ordered from largest to smallest cumulative values.
- The chart legend shows the individual accounts. Click an account to show/hide values for that account.
- The chart and legend show up to 20 accounts and, if needed, an additional item labeled 'Others' that represents all accounts that are not in the top 20.

## Show All

Click **Show All** at the bottom of the chart to view and download data in tabular format.

## Investments and Savings Charts

Actions for your cloud workloads usually have cost savings or investments attached to them. For example, deleting unattached volumes can lower your costs significantly (savings), while scaling a VM to a different

tier to improve performance could incur additional costs (investments). These charts highlight total savings and investments as a result of executing actions. Turbonomic uses billing data from your cloud provider to calculate savings and investments.

You can add these charts to the default views in the Home Page or to your custom dashboards.

Points to consider:

- Billing data from your cloud provider generally takes 1 to 2 days to update fully. Savings and investment figures may change until the data is fully updated.
- The charts show savings and investments data for actions executed *after* the installation of version 8.9.3. Data for actions executed before this version will not be shown, but can be extracted from the platform using a script. If you need to extract the data, contact your Turbonomic representative for assistance.

## Scale Actions

Scale actions involve moving an entity to a different compute tier or adjusting the entity's allocated capacity. Currently, Turbonomic calculates savings and investments when scaling the following entities:

- VMs (AWS, Azure, and GCP)
- Volumes (AWS and Azure)
- Databases (Azure SQL vCore or DTU)
- Virtual Machine Specs (Azure App Service plans)

Turbonomic calculates savings and investment *per entity*. Calculation is based on the *before-action* and *after-action* costs.

- Before-action cost is the cost of an entity before an action was executed. This is based on the cost snapshot on the day of action execution.
- After-action cost is the cost of an entity as reflected in the daily billing report from the cloud provider.

### Note:

When the result of the calculation is a negative amount, Turbonomic considers savings or investments to be \$0.

Turbonomic reads the daily billing report from the cloud provider to determine the current daily cost of an entity. Calculation also takes into account workload uptime (for VMs).

Points to consider:

- Calculation can adjust to varying VM uptime and discount coverage over time.
- Calculation stops for terminated entities or entities that Turbonomic no longer discovers.
- Calculation stops if there is a cost change in the entity that is *not* the result of a Turbonomic action.


## Delete Actions

Delete actions always result in savings. Currently, Turbonomic calculates savings when deleting the following entities:

- Unattached volumes (AWS, Azure, and GCP)
- Empty Virtual Machine Specs (Azure App Service plans)

Turbonomic checks the daily cost of an entity on the day the entity was deleted. The daily bill from the cloud provider is not used to calculate savings.

## Chart Settings

Click the More options icon (  ), and then select **Edit** to modify the following settings:

- Scope



By default, the charts are scoped to your global environment. You can change the scope to one or several accounts, billing families, groups, or entities.

- Timeframe
  - Last 7 Days or Last 30 Days – Each data point in the chart shows total savings or investments for the given day.
  - Last Year – Each data point in the chart shows total savings or investments for the given month.

- Chart Type

You can set the display to:

- Text and Bar Chart
- Text and Area Chart
- Stacked Bar Chart
- Area Chart
- Text

- Type

Switch between the **Savings** and **Investments** views. You can also change the displayed data to **Cumulative Savings** or **Cumulative Investments** to see how the savings or investment costs accumulate over time.

- Group By

This setting breaks down data points by the selected group. By default, no group is selected. Each data point only shows the total values for a given day or month, depending on the selected timeframe.

The following groups are supported:

- Account – Cloud accounts discovered in your environment
- Action Type – Scale and delete actions
- Cloud Service Provider – AWS, Azure and GCP
- Entity Type – Virtual Machine, Volume, Database (Azure only), Virtual Machine Spec (Azure only)
- Region – Regions for the entities
- Resource Group (Azure only) – Available resource groups in your environment

For example, if you selected **Account**:

- The chart represents each account as a distinct color.
- When you hover on a data point, the chart shows a list of accounts, ordered from largest to smallest values.
- The chart legend shows the individual accounts. You can click an account to show/hide values for that account.
- The chart and legend show up to 20 accounts and, if needed, an additional item labeled 'Others' that represents all accounts that are not in the top 20.

## Show All

Click **Show All** at the bottom of the chart to view and download data in tabular format.

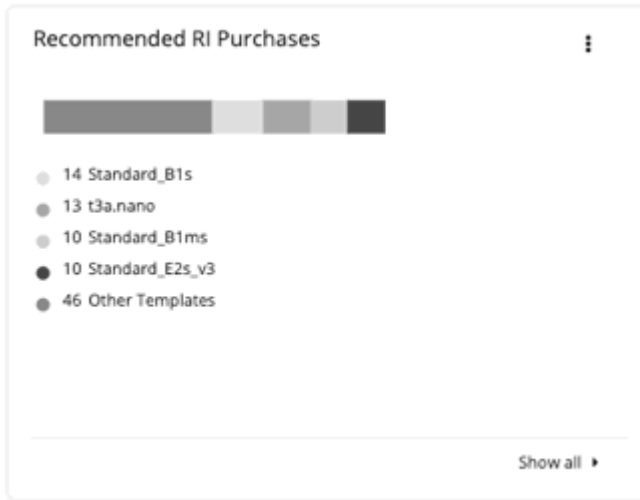
## Recommended RI Purchases Charts

Turbonomic can recommend purchasing instance types at a discounted rate to help you increase the percentage of VMs covered by discounted pricing and reduce on-demand costs. This chart shows your pending purchases. Download the list of purchases and then send it your cloud provider or representative to initiate the purchase process.

### Note:

Purchase actions should be taken along with the related VM scaling actions. To purchase discounts for VMs at their current sizes, run a [Buy VM Reservation Plan](#).

Currently, Turbonomic can recommend purchase actions for AWS and Azure. Purchase actions for GCP will be introduced in a future release.



## Factors Affecting Recommendations

To identify VMs that are good candidates for discounted pricing, Turbonomic analysis considers the history of a VM (by default, the last 21 days), and it looks for:

- Activity

If the VM's VCPU utilization percentile is 20% or higher, then Turbonomic considers it an active VM.

- Stability

If there have been no start, stop, or resize actions for the VM for 16 of the last 21 days, then Turbonomic considers it stable.

If the current discount inventory cannot support the VM, or if supporting it would exceed your desired coverage, then Turbonomic can recommend purchasing additional discounts.

Turbonomic generates purchase actions on a two-week cycle. It also generates a new set of actions if the discount inventory changes or after the platform restarts.

Be aware of the following:

- Different types of discounts have different costs, so the choice between using on-demand or discounted pricing can vary depending on your [purchase profile](#).
- Turbonomic can only estimate costs because the full data is only available after you complete the purchase. Estimates reflect costs you would see after scaling workloads to the newly purchased instance types. For scaling to already-purchased instance types, the chart reflects the actual costs.
- As Turbonomic generates purchase actions, it assumes that any other pending actions for the workload will also be executed. For example, assume a workload running on an r4.xlarge template. If Turbonomic recommends changing that instance type to an m5.medium, it can recommend that you purchase a discounted m5 to cover the workload and reduce costs. This purchase could be on a region that currently doesn't have any m5 workloads. The purchase recommendation assumes you will move the workload to that other region.
- For AWS RIs:
  - For environments that use the *Instance Size Flexible* rules, Turbonomic can recommend that you buy multiple RIs of smaller instance types to cover the resource requirements of larger instance types.

For example, rather than buying one t2.small RI, Turbonomic can recommend that you buy four t2.nano RIs to offer an equivalent discount.

- For environments that consolidate billing into Billing Families, Turbonomic recommends purchases that are within the given billing family. For more information, see [AWS Billing Families](#).

## Show All

Click **Show All** to see a table with details for each discount.

The table shows the properties, up-front cost, and break-even period for each discount. The break-even period is the time at which savings will exceed the up-front cost, rounded to the month. The Cost Impact column indicates the monthly savings you would realize when you buy a specific discount.

When you choose one or more check boxes, the total count, up-front cost, and savings appear at the top.

| AWS                                         |          | AZURE            |       |          |        |             |                     |               |                   |                 |             |                 |   |
|---------------------------------------------|----------|------------------|-------|----------|--------|-------------|---------------------|---------------|-------------------|-----------------|-------------|-----------------|---|
| <b>Buy Actions</b> 20                       |          | Savings \$799/mo |       |          |        |             |                     |               |                   |                 |             | EXECUTE ACTIONS | ↓ |
| <input type="text" value="Type to search"/> |          |                  |       |          |        |             |                     |               |                   |                 |             | ADD FILTER      |   |
| <input type="checkbox"/>                    | Account  | Instance Type    | Count | Platform | Term   | Payment     | Region              | Up-Front Cost | Break Even Period | Action Category | Cost Impact | Action          |   |
| <input type="checkbox"/>                    | Advanced | r5a.large        | 8     | Linux    | 1 Year | All Upfront | aws-US West (N. C.) | \$5,192       | 8 months          | SAVINGS         | ↓ \$210/mo  | DETAILS         |   |
| <input type="checkbox"/>                    | Advanced | c5a.large        | 6     | Linux    | 1 Year | All Upfront | aws-US West (N. C.) | \$2,934       | 7 months          | SAVINGS         | ↓ \$171/mo  | DETAILS         |   |
| <input type="checkbox"/>                    | Advanced | r5a.large        | 5     | Linux    | 1 Year | All Upfront | aws-US East (Ohio)  | \$2,910       | 8 months          | SAVINGS         | ↓ \$133/mo  | DETAILS         |   |

| AWS                                         |              | AZURE            |         |        |                     |               |                   |                 |             |         |  |                 |   |
|---------------------------------------------|--------------|------------------|---------|--------|---------------------|---------------|-------------------|-----------------|-------------|---------|--|-----------------|---|
| <b>Buy Actions</b> 27                       |              | Savings \$928/mo |         |        |                     |               |                   |                 |             |         |  | EXECUTE ACTIONS | ↓ |
| <input type="text" value="Type to search"/> |              |                  |         |        |                     |               |                   |                 |             |         |  | ADD FILTER      |   |
| <input type="checkbox"/>                    | Subscription | Product Name     | Quan... | Term   | Region              | Up-Front Cost | Break Even Period | Action Category | Cost Impact | Action  |  |                 |   |
| <input type="checkbox"/>                    | Dev          | Standard_D2s_v3  | 6       | 1 Year | azure-North Central | \$3,042       | 7 months          | SAVINGS         | ↓ \$173/mo  | DETAILS |  |                 |   |
| <input type="checkbox"/>                    | Dev          | Standard_B1ls    | 49      | 1 Year | azure-East US       | \$1,323       | 7 months          | SAVINGS         | ↓ \$73/mo   | DETAILS |  |                 |   |
| <input type="checkbox"/>                    | Dev          | Standard_DS1_v2  | 2       | 1 Year | azure-Brazil South  | \$798         | 6 months          | SAVINGS         | ↓ \$57/mo   | DETAILS |  |                 |   |

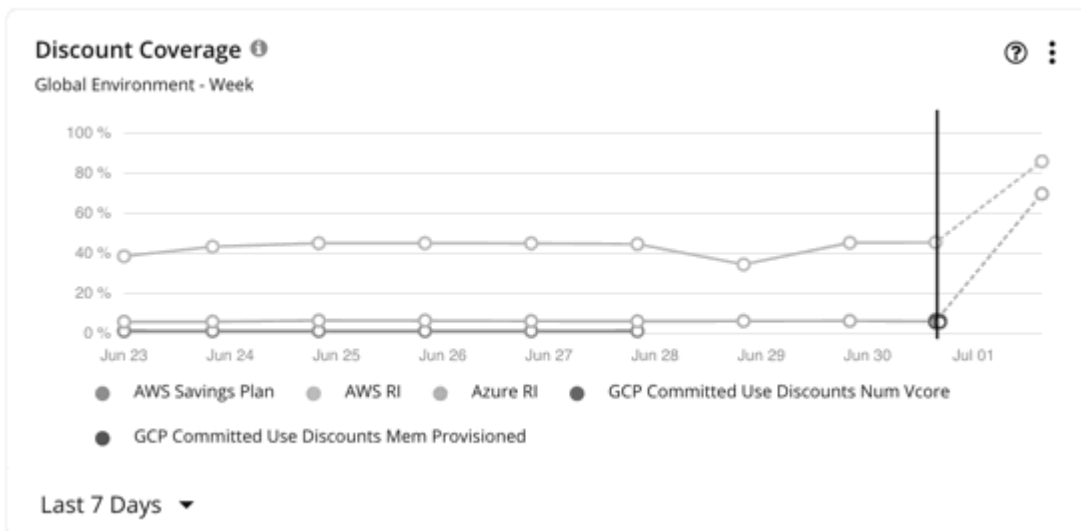
## Chart Type

You can set the display to:

- Text
- Ring Chart
- Horizontal Bar

## Discount Coverage Chart

This chart shows the percentage of VMs covered by discounts. If you have a high percentage of on-demand VMs, you should be able to reduce your monthly costs by increasing coverage. To increase coverage, you scale VMs to instance types that have existing capacity.



To identify VMs that are good candidates for discounted pricing, Turbonomic analysis considers the history of a VM (by default, the last 21 days), and it looks for:

- Activity  
If the VM's VCPU utilization percentile is 20% or higher, then Turbonomic considers it an active VM.
- Stability  
If there have been no start, stop, or resize actions for the VM for 16 of the last 21 days, then Turbonomic considers it stable.

If the current discount inventory cannot support the VM, or if supporting it would exceed your desired coverage, then Turbonomic can recommend purchasing additional discounts.

## AWS RIs

If you set the scope to a specific AWS account, the chart shows the RI coverage for the workloads for the account, plus any RIs for the billing family.

The data point on the solid vertical line shows the latest data that was polled from your environment. Data points to the left of the vertical line represent historical data, while data points to the right are projections into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of coverage, based on normalization factors

Normalization factor is a measure of RI capacity that you can use to compare or combine the capacity for different instance families.

Turbonomic measures RI coverage in terms of normalization factors. It compares the number of RIs calculated as normalization factors that cover workload capacity with the total number of normalization factors for a given Turbonomic scope. Each workload is assigned normalized units depending on its instance type.

### Note:

In AWS, you can turn off RI discount sharing for specific accounts. These accounts will not share any discounts with other accounts. Turbonomic does not recognize RI coverage or utilization for these accounts. For example, the RI Coverage and RI Utilization charts will show zero values.

## AWS Savings Plans

To view data for AWS Savings Plans, you must:

- Set the scope to your global environment.
- Choose a timeframe that shows daily or monthly data points (Last 7 Days, Last 30 Days, or Last Year)

AWS measures your Savings Plans commitment in \$/hour, but shows daily and monthly costs in Cost Explorer. Turbonomic polls Cost Explorer periodically to obtain the latest cost data, and then uses that data to calculate Savings Plans utilization or coverage *per day*. For this reason, you will not see Savings Plans data if you choose a timeframe that shows hourly data points (Last 2 Hours or Last 24 Hours).

### Note:

AWS timestamps data in UTC, but the chart presents data in your local time. This difference could result in the appearance of a missing day in the chart, but has no effect on data completeness (the chart always reflects the complete data set).

The chart shows historical data, represented by data points to the left of the solid vertical line. Data for the current day is not available since the latest data that AWS provides is always a few days old. In addition, Turbonomic does not project coverage into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of coverage

## Azure Reservations

If you set the scope to a specific Azure subscription, this chart shows the reservation coverage for the workloads for the subscription, plus any shared reservations and single-scope reservations owned by this subscription.

The data point on the solid vertical line shows the latest data that was polled from your environment. Data points to the left of the vertical line represent historical data, while data points to the right are projections into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of coverage, based on ratios.

Ratio refers to the number of Azure reservation units that cover workload capacity compared to the total number of reservation units for a given Turbonomic scope. Each workload is assigned reservation units based on its instance type.

## GCP Committed Use Discounts

The chart displays the latest data that was polled from your environment, but does *not* show historical data or project coverage into the future.

Hover on a data point in the chart to see the following information:

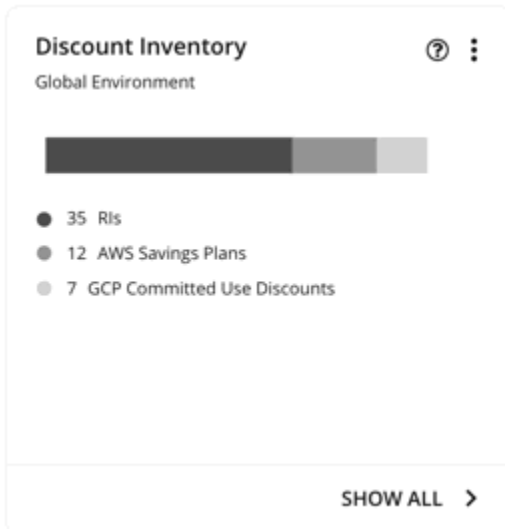
- The date and time for the data point
- The percentage of coverage

### ***Discount Inventory Chart***

This chart lists the cloud provider discounts discovered in your environment.

- AWS Reserved Instances (RIs) and Savings Plans for regular and [GovCloud](#) workloads
- Azure reservations for regular and [Azure Government](#) workloads
- GCP committed use discounts

## Chart Type



You can set the display to:

- Text
- Ring Chart
- Horizontal Bar

## Show All

Click **Show All** at the bottom of the chart to see detailed information for the discounts in scope. If your scope includes multiple cloud providers, each provider will have its own tab.

| Discount Inventory       |                |               |                   |       |               |                          |         |               |                             |                     |              |        |              |                | ✕         |
|--------------------------|----------------|---------------|-------------------|-------|---------------|--------------------------|---------|---------------|-----------------------------|---------------------|--------------|--------|--------------|----------------|-----------|
| AWS RESERVED INSTANCES   |                |               | AWS SAVINGS PLANS |       |               | AZURE RESERVED INSTANCES |         |               | GCP COMMITTED USE DISCOUNTS |                     |              |        |              |                |           |
| <input type="checkbox"/> |                | Total         |                   | 8     | Count         | 12                       | Cost    | \$1,790.83/mo | Savings                     | \$1,434.16/mo       |              |        |              |                |           |
| <input type="checkbox"/> | Reservation ID | Account       | Instance Type     | Count | Location      | Platform                 | Tenancy | Class         | Payment                     | Current Utilization | Est. Savings | Term   | Exp. Date    | Effective Cost | Li... VMs |
| <input type="checkbox"/> |                | Quality Engin | t3.nano           | 1     | aws-ap-northe | Linux                    | Default | Convertible   | All Upfront                 | 0%                  | \$0.00/mo    | 1 Year | Nov 02, 2022 | \$3.67/mo      | 0 VM      |
| <input type="checkbox"/> |                | Quality Engin | t3.nano           | 1     | aws-ap-northe | Linux                    | Default | Convertible   | All Upfront                 | 0%                  | \$0.00/mo    | 1 Year | Nov 02, 2022 | \$3.67/mo      | 0 VM      |
| <input type="checkbox"/> |                | Development   | t3.micro          | 1     | aws-US East ( | Linux                    | Default | Convertible   | No Upfront                  | 100%                | \$2.12/mo    | 1 Year | Feb 28, 2023 | \$5.48/mo      | 1 VM      |

Each row in the table corresponds to a discount. Note that there can be several discounts for an Azure subscription or AWS/GCP account, and each discount displays as its own row. Table columns show basic information obtained from the cloud provider, such as the name/ID of the discount, the subscription/account that uses that discount, instance type and location, term, and expiration dates. Click a subscription/account to narrow the scope.

The table supports the following general functionality:

- **Totals:** At the top of the page, you will see the total number of discovered discounts. For AWS RIs and Azure reservations, you will also see total costs and savings. As you select one or more checkboxes, the information changes to reflect the totals for your selections.
- **Column Sorting:** Click any column heading to sort the list.
- **Download:** Click the Download icon at the upper right section of the page to download the table as a CSV file.

## Azure Reservations and AWS RIs

When you add an Azure EA account or an AWS master account as your primary cloud target, Turbonomic gains full insight into the discounts for your billing families. Even as you selectively add Azure subscriptions or AWS member accounts as secondary targets, Turbonomic remains aware of all discounts and how they are utilized across the board, and can thus recommend more accurate discount optimization and purchase actions.

### Points to consider:

- For AWS, if you added some member accounts as targets, but not a master account, Turbonomic will not reflect discounts for member accounts that you have not added as targets.
- For Azure:
  - It could take Turbonomic up to a day to discover newly purchased Azure reservations.
  - There can be delays in billing information updates that Azure makes available to Turbonomic. If this happens, analysis might use partial billing data in its calculations and show incomplete costs for discounts from non-added Azure subscriptions.

Set the scope to your global environment to view the full inventory. When you click **Show All** at the bottom of the chart, pay attention to the following information shown in the table:

- For discounts in *added* accounts (Azure subscriptions or AWS member accounts):

| Reservation ID | Subscription | Name                         | Product Name    | Quantity | Location        | Scope  | Current Utilization | Est. Savings | Term   | Exp. Date    | Effective Cost | Linked VMs |
|----------------|--------------|------------------------------|-----------------|----------|-----------------|--------|---------------------|--------------|--------|--------------|----------------|------------|
|                | EA - Dev     | Standard_DS2_v2_renewed      | Standard_DS...  | 1        | azure-East US 2 | Shared | 100%                | \$29.14/mo   | 1 Year | Aug 24, 2022 | \$54.08/mo     | 1 VM*      |
|                | EA - Dev     | Standard_DS2_v2_renewed      | Standard_DS...  | 1        | azure-East US 2 | Shared | 100%                | \$29.14/mo   | 1 Year | Sep 03, 2022 | \$54.08/mo     | 1 VM*      |
|                | EA - Dev     | Standard_D2as_v4_renewed     | Standard_D2a... | 1        | azure-East US 2 | Shared | 100%                | \$28.33/mo   | 1 Year | Sep 03, 2022 | \$41.75/mo     | 1 VM       |
|                | EA - Dev     | Standard_F1s_renamed_renewed | Standard_F1s    | 1        | azure-East US 2 | Shared | 100%                | \$9.20/mo    | 1 Year | Sep 03, 2022 | \$27.08/mo     | 1 VM*      |

- **Subscription** column (for Azure) or **Account** column (for AWS)

This column shows the account name for the discount. Click the name to set the scope to that account. Note that there can be several discounts for an account, and each discount displays as its own row.

### Note:

If there is a failure to re-validate the account for some reason, Turbonomic shows it as a *non-added* account in the Discount Inventory page.

- **Current Utilization** column

This column shows the percentage of discount capacity currently used by VMs in all accounts. Turbonomic estimates the percentage if there are VMs in non-added accounts that use the discount (since the exact number of VMs is unknown).

- **Linked VMs** column

This column shows how many VMs in the account use the discount. Click the number to view the VMs.

A star symbol (\*) after the number indicates that there are VMs from *non-added* accounts that also use the discount. Since you have not added those accounts, the exact number of VMs is unknown.

- For discounts in *non-added* accounts:

| Reservation ID | Subscription | Name             | Product Name    | Quantity | Location        | Scope        | Current Utilization | Est. Savings | Term   | Exp. Date    | Effective Cost | Linked VMs |
|----------------|--------------|------------------|-----------------|----------|-----------------|--------------|---------------------|--------------|--------|--------------|----------------|------------|
|                | 9224a        | Standard_F1_r... | Standard_F1     | 1        | azure-East US 2 | Shared       | 0%                  | \$0.00/mo    | 1 Year | Aug 24, 2022 | \$27.08/mo     | 0 VM       |
|                | 9224a        | Standard_F1s_... | Standard_F1s    | 1        | azure-East US 2 | Shared       | 0%                  | \$0.00/mo    | 1 Year | Aug 24, 2022 | \$27.08/mo     | 0 VM       |
|                | acbc8        | Standard_D2s_... | Standard_D2a_v4 | 2        | azure-East US 2 | acbc86dd-2dk | 38%                 | \$21.25/mo   | 1 Year | Aug 24, 2022 | \$83.50/mo     | –          |

- **Account** column (for AWS) or **Subscription** column (for Azure)

This column shows a grayed-out, non-clickable name to indicate that you have not added the account as a target. Turbonomic is aware of this account and if the given discount is shared with other accounts because you have added a master or EA account.

– **Current Utilization** column

This column shows the percentage of discount capacity currently used by VMs in all accounts. Turbonomic estimates the percentage if there are VMs in non-added accounts that use the discount (since the exact number of VMs is unknown).

– **Linked VMs**

If the number under this column is 1 or more, the number indicates VMs from *added accounts* that use the discount. Click the number to view the VMs.

If the number is 0 (zero), then the discount is currently not used anywhere.

A hyphen symbol (-) indicates that there are VMs from other *non-added* accounts that also use the discount. Since you have not added those accounts, the exact number of VMs is unknown.

## AWS Savings Plans

If you added targets that are AWS accounts with read-only access to the AWS Savings Plans API, Turbonomic uses this chart to present the Savings Plans that it discovered in your cloud environment (including [GovCloud](#)) and the instance types they use.

| <input type="checkbox"/> Savings Plan ID | Account ⓘ | Type    | Payment     | Instance Family | Location | Commitment | Term    | Start Date   | Exp. Date    |
|------------------------------------------|-----------|---------|-------------|-----------------|----------|------------|---------|--------------|--------------|
| <input type="checkbox"/> 55555 ...       | Prod      | Compute | All Upfront | All             | All      | \$0.001/hr | 1 Year  | Dec 28, 2020 | Dec 28, 2021 |
| <input type="checkbox"/> 45555 ...       | Prod      | EC2     | No Upfront  | t3              | aws-...  | \$0.001/hr | 3 Years | Dec 29, 2020 | Dec 29, 2023 |

## GCP Committed Use Discounts

Turbonomic discovers committed use discounts for your workloads when you add the following as targets:

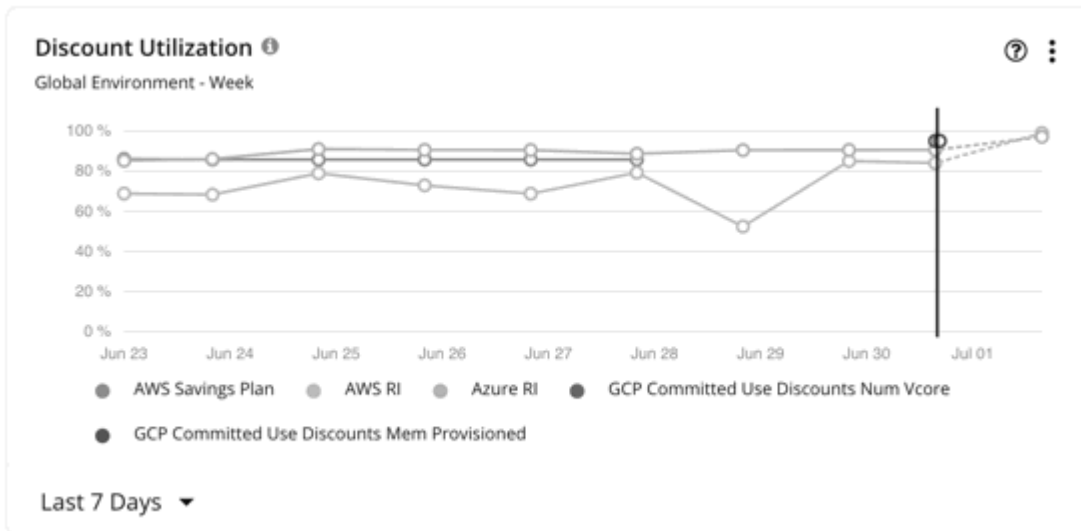
- A service account with the "Billing Account Viewer" role to the related billing accounts
- Billing accounts

| <input type="checkbox"/> Name         | Account ⓘ | Status | Region               | Type          | Payment     | Instance Family | Cores | Memory | Term    | Start Date   | End Date     |
|---------------------------------------|-----------|--------|----------------------|---------------|-------------|-----------------|-------|--------|---------|--------------|--------------|
| <input type="checkbox"/> commitment-1 |           | Active | us-west4             | Family Scoped | All upfront | N2              | 1     | 4 GB   | 3 Years | Oct 15, 2021 | Oct 15, 2024 |
| <input type="checkbox"/> commitment-1 |           | Active | europa-north1        | Family Scoped | All upfront | N2              | 1     | 2 GB   | 1 Year  | Oct 21, 2021 | Oct 21, 2022 |
| <input type="checkbox"/> commitment-1 |           | Active | northamerica-northea | Family Scoped | All upfront | N2D             | N/A   | 2 GB   | 1 Year  | Jan 27, 2022 | Jan 27, 2023 |
| <input type="checkbox"/> commitment-2 |           | Active | us-central1          | Family Scoped | All upfront | E2              | N/A   | 4 GB   | 1 Year  | Oct 21, 2021 | Oct 21, 2022 |

## Discount Utilization Chart

This chart shows how well you have utilized your current discount inventory. The desired goal is to maximize the utilization of your inventory and thus take full advantage of the discounted pricing offered by your cloud provider.





## AWS RIs

You can set the scope to your global cloud environment or to individual accounts, billing families, or regions. Scoping to an account shows the RI utilization for the workloads for the entire billing family.

### Note:

Under very rare circumstances, you can have RIs on payment plans that do not resolve to 1-year or 3-year terms. In this case, AWS does not return pricing data for those RIs. Turbonomic does not include such RIs in its calculations of RI utilization or RI cost.

The data point on the solid vertical line shows the latest data that was polled from your environment. Data points to the left of the vertical line represent historical data, while data points to the right are projections into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of utilization, based on normalization factors

Normalization factor is a measure of RI capacity that you can use to compare or combine the capacity for different instance families.

Turbonomic measures RI coverage in terms of normalization factors. It compares the number of RIs calculated as normalization factors that cover workload capacity with the total number of normalization factors for a given Turbonomic scope. Each workload is assigned normalized units depending on its instance type.

### Note:

In AWS, you can turn off RI discount sharing for specific accounts. These accounts will not share any discounts with other accounts. Turbonomic does not recognize RI coverage or utilization for these accounts. For example, the RI Coverage and RI Utilization charts will show zero values.

## AWS Savings Plans

To view data for AWS Savings Plans, you must:

- Set the scope to your global environment.
- Choose a timeframe that shows daily or monthly data points (Last 7 Days, Last 30 Days, or Last Year)

AWS measures your Savings Plans commitment in \$/hour, but shows daily and monthly costs in Cost Explorer. Turbonomic polls Cost Explorer periodically to obtain the latest cost data, and then uses that

data to calculate Savings Plans utilization or coverage *per day*. For this reason, you will not see Savings Plans data if you choose a timeframe that shows hourly data points (Last 2 Hours or Last 24 Hours).

**Note:**

AWS timestamps data in UTC, but the chart presents data in your local time. This difference could result in the appearance of a missing day in the chart, but has no effect on data completeness (the chart always reflects the complete data set).

The chart shows historical data, represented by data points to the left of the solid vertical line. Data for the current day is not available since the latest data that AWS provides is always a few days old. In addition, Turbonomic does not project coverage into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of utilization, based on the total utilized and committed costs per day

## Azure Reservations

You can set the scope to your global cloud environment or to individual subscriptions, billing families, or regions. Scoping to a subscription shows the reservations utilization for workloads for the entire billing family or for single and shared subscriptions.

The data point on the solid vertical line shows the latest data that was polled from your environment. Data points to the left of the vertical line represent historical data, while data points to the right are projections into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of utilization, based on ratios.

Ratio refers to the number of Azure reservation units that cover workload capacity compared to the total number of reservation units for a given Turbonomic scope. Each workload is assigned reservation units based on its instance type.

## GCP Committed Use Discounts

The chart displays the latest data that was polled from your environment, but does *not* show historical data or project utilization into the future.

Hover on a data point in the chart to see the following information:

- The date and time for the data point
- The percentage of utilization

## Cloud Estimated Cost Charts

Cloud Estimated Cost charts show estimated monthly costs and investments for the cloud. Monthly cost amounts are summarized as amounts with and without actions.

## Display

The chart shows the information as a Text chart.

## Volume Summary Charts

To help you manage your costs on the public cloud, these charts show the distribution and costs of volumes for the given scope. In this way, you can see how volume utilization affects your costs. For these charts, Turbonomic calculates the costs based on the cost information from the cloud targets.

These charts show the following data:

- Tiers

The chart breaks down volumes by tier (disk type) and shows the volume count and monthly cost for each tier.

- Volume State

The chart breaks down volumes by attachment state (attached or unattached) and shows the volume count and monthly cost for each state. For unattached volumes, you can reduce your cloud cost by the given amount if you delete these volumes. Click **Show All** and then click the **Details** button for an unattached volume to execute a delete action.

**Note:**

For an Optimize Cloud plan, the Volume Tier Summary chart shows 'Current' and 'Optimized' results. The 'Current' result includes currently unattached volumes that you can delete to reduce costs, while the 'Optimized' result assumes that unattached volumes have been deleted. To see a list of unattached volumes, click **Show Changes** at the bottom of the chart. For details about Optimize Cloud plans, see [Optimize Cloud Plan Results](#).

For a detailed breakdown, click **Show All** at the bottom of the chart. If you have multiple cloud providers, each provider will have its own tab. Click any column heading to sort the list. When you choose one or more check boxes, the total appears at the top.

**Note:**

For Azure environments with VMs in Scale Sets, for any VMs that are powered off, the associated volume shows a utilization of zero GB. This is an accurate presentation of the data that the Azure environment returns for such a powered-off VM. However, it is likely that some of the volume capacity is currently utilized.

## Chart Unit

If you are scoped to a particular cloud provider, you can sort tiers and volumes by clicking the Edit option at the top-right corner of the chart, and then choosing one of the following units:

- **Count** – Sort by volume count, from largest to smallest.
- **Cost** – Sort by monthly cost, from highest to lowest.

## On-Prem Chart Types

These chart widgets provide information on the status of your on-prem environment.

### *Density Charts*

Density charts show the number of resource consumers (virtual machines or containers) per provider (host or storage). If available, choose the **Show Density** checkbox to see the ratio of consumers to providers.

These charts also show the desired count of virtual machines, assuming you want to fill the headroom completely. Note that the Desired Workloads values are the results of running plans. These plans can calculate workload moves within a cluster to gain more efficiency, but they always respect the cluster boundaries – the plans never move VMs to hosts on different clusters.

To display relevant data, you must set the scope to your global environment or a cluster group. Other scopes are not supported.

To display relevant data, you must set the scope to your global environment or a cluster group. Other scopes are not supported.

## Chart Type

You can set the display to:

- Stacked Bar Chart

- Line Chart

### ***Ports Charts***

Ports charts show the most utilized northbound or southbound ports in your on-prem environment over a given time period. These charts are useful in Fabric environments where you license port channels.

### **Display**

The chart shows the information as Tabular.

### ***Headroom Charts***

Headroom charts show how much extra capacity your clusters have to host workloads.

To calculate cluster capacity and headroom, Turbonomic runs nightly plans that take into account the conditions in your current environment. The plans use the Economic Scheduling Engine to identify the optimal workload distribution for your clusters. This can include moving your current VMs to other hosts within the given cluster, if such moves would result in a more desirable workload distribution. The result of the plan is a calculation of how many more VMs the cluster can support.

To calculate VM headroom, the plan simulates adding VMs to your cluster. The plan assumes a certain capacity for these VMs, based on a specific VM template. For this reason, the count of VMs given for the headroom is an approximation based on that VM template.

You can specify the following types of Headroom charts:

- CPU Headroom
- Memory Headroom
- Storage Headroom

### **Commodity**

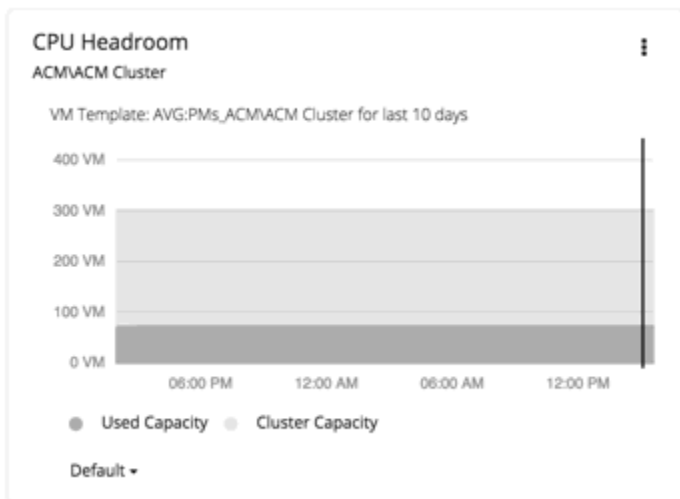
You can choose:

- CPU Headroom
- Memory Headroom
- Storage Headroom

### **Display**

The chart shows the information as an Area chart.

Example:



### Exhaustion Time Chart

This chart shows the current growth of workloads and projects when workloads will exceed the capacity of your current infrastructure. This is useful for future planning (for example, if you might need to buy more hardware).

You can track CPU, memory, and storage as well as the average monthly Virtual Machine growth and the average VM template. The amount of time is presented as days. For example, storage will be used up in 41 days.

### Display

The chart shows the information as a Text chart.

## Creating Groups

Groups assemble collections of resources for Turbonomic to monitor and manage. When setting scope for your Turbonomic session, you can select groups to focus on those specific resources. For example, if you have a number of VMs devoted to a single customer, you can create a group of just those VMs. When running a planning scenario you can set the scope to work with just that group.

Turbonomic discovers groups that exist in your environment. These groups include PM clusters, and entities grouped by different logical boundaries. For example, Turbonomic discovers Storage by Disk Array, Physical Machines by Datacenter, and VMs by Network. In addition, Turbonomic discovers pools such as virtual datacenters, or folders that implement specific HA policies.

You can also create custom groups. Turbonomic supports two custom-grouping methods:

- **Dynamic** — You define these groups by specific criteria. You can group services according to naming conventions (all VM names that start with **ny**), resource characteristics (all physical machines with four CPUs), or other criteria such as time zone or number of CPUs.

These groups are dynamic because Turbonomic updates the group as conditions change.

- **Static** — You create these groups by selecting the specific group members.

### Note:

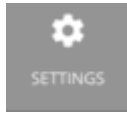
You should never use the Turbonomic user interface to delete discovered groups. If you do, later analysis cycles will discover them again, and add them to your environment. But until it recreates those groups, any analysis that relies on those groups can give unexpected results.

You *can* delete any custom group you have created. Before you do, you should verify that you do not have any charts, plans, or policies that use the group you want to delete to set their scopes. After you delete

the group, such charts, plans, or policies will lose their scope. For example, a policy with no scope has no effect.

To create a group:

1. Navigate to the Settings Page.

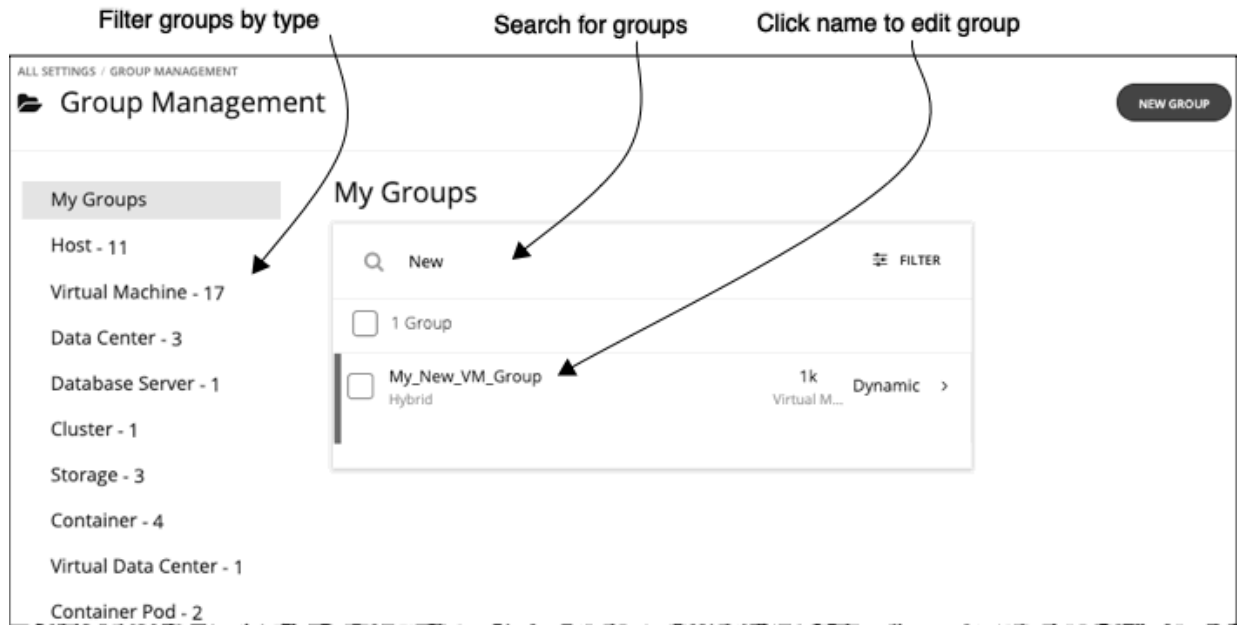


Click to navigate to the Settings Page. From there, you can perform a variety of Turbonomic configuration tasks.

2. Choose **Groups**.



Click to navigate to the Group Management Page.



This page lists all the custom groups that you currently have configured for Turbonomic. You can:

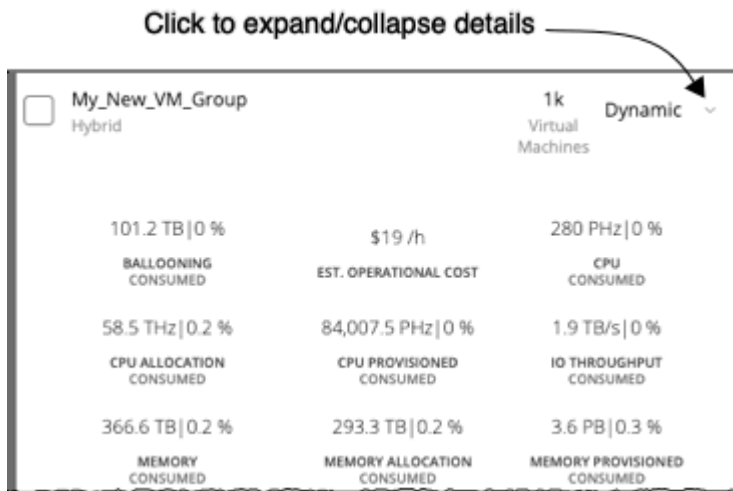
- Expand an entry to see group details
- Select an entry to delete the group
- Click a group name to edit it

For a dynamic group, you can edit the set of criteria that select the group members. For a static group, you can add or subtract specific members.

- Create new groups

To work with a long list of groups, you can filter by group type. For example, only show groups of VMs, or groups of host machines. You can also type a string in the **Search** field to filter the list.

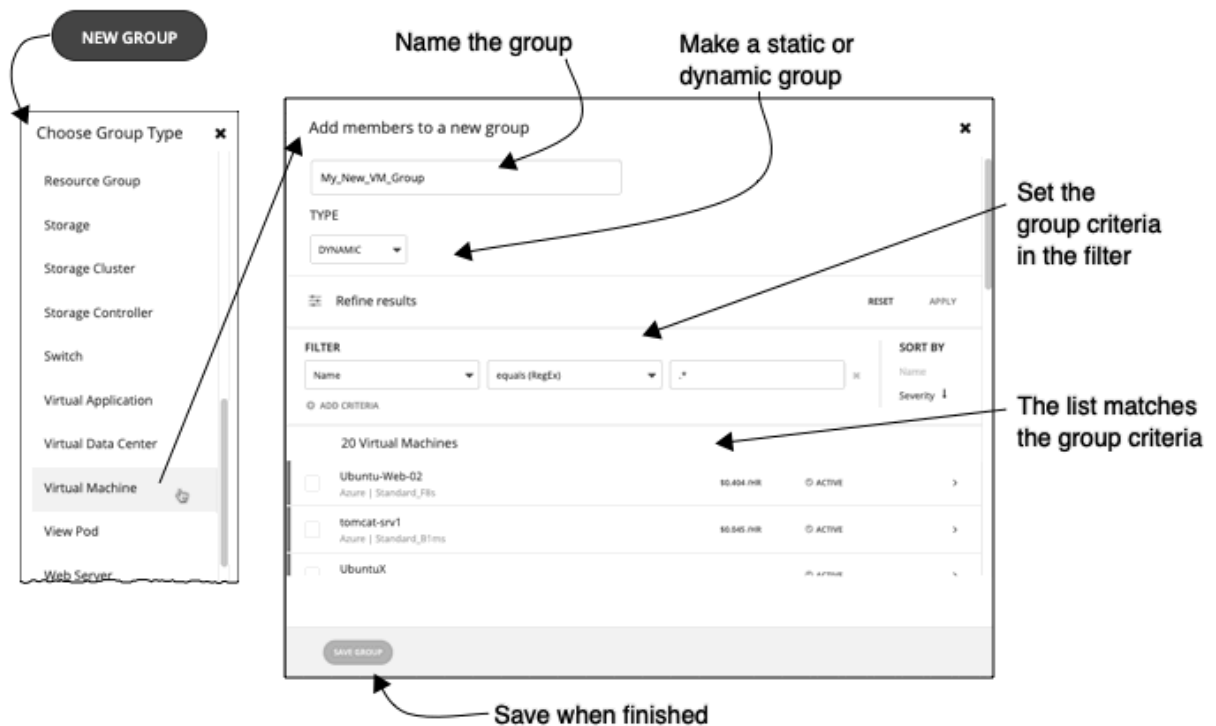
3. Expand an entry to see group details.



The details show you information about related entities such as how many hosts provide resources for a group of VMs. If there are any pending actions for the group, the details list those actions as well.

4. Create a new group.

Click **NEW GROUP**.



Next, choose a group type.

Then, specify the group settings:

- Give the group a unique name. To prevent issues, you should never use duplicate names for groups of the same entity type.
- Set whether the group will be static or dynamic.

To create a static group, select the member entities from the list. To filter the list, set group criteria.

To create a dynamic group, set group criteria. The list updates to show the resulting group members.

- Specify group criteria.

These criteria are entity attributes that determine group membership. You might create a group of all VMs that have 4 VCPUs. You can choose properties of the member entities, and you can choose properties of entities that are related to the members. For example, you can make a group of VMs that are hosted by PMs with the substring "Development" in their names.

As you set criteria, the list of entities updates to show the member entities. You also can sort the list by severity (per the most critical entity in group) or group name.

Note that you can use regular expression to express your match strings.

- When you are finished, save the group.

**Save** adds this group to the **My Groups** collection.

## Working With Policies

---

Policies set business rules to control how Turbonomic analyzes resource allocation, displays resource status, and recommends or executes actions. Turbonomic includes two fundamental types of policies.

- Placement Policy

A placement policy is a set of rules that Turbonomic must satisfy when generating placement recommendations for on-prem and containerized workloads.

For details, see [Placement Policies](#).

- Automation Policy

An automation policy is a set of rules that Turbonomic must satisfy when executing [non-parking actions](#) on public cloud or on-prem workloads, or changing settings that affect analysis and action generation.

For details, see [Automation Policies](#).

### Policy Management

Use the Policy Management page to view and manage policies. The page shows the following categories of policies:

- All Policies – All the currently defined policies
- Placement Policies
  - Imported Placement Policies – Placement policies [discovered](#) from your targets
  - Turbonomic Segments – Placement policies [created](#) from the Policy Management page
- Automation Policies
  - Imported Automation Policies – Automation policies [discovered](#) from your targets
  - User Defined Automation Policies – Automation policies [created](#) from the Policy Management page
  - Defaults – [Default](#) automation policies for the different entity types

#### Note:

To see the policies that are applied by a particular entity, set the scope to that entity (from Search or the supply chain), and then click the Policies tab. For more information, see [Scope Policies](#).

## Placement Policies

A placement policy is a set of rules that Turbonomic must satisfy when generating placement recommendations for on-prem and containerized workloads.

With these policies, Turbonomic can recommend placement actions that comply with your business rules. For example, a placement policy can constrain VMs to specific hosts. If a VM needs to move to a different host due to resource congestion, Turbonomic will generate an action to move the VM to one of the hosts defined in the policy.



Turbonomic discovers placement policies from your targets, and allows you to create your own policies.

**Note:**

You can enable or disable any placement policy to affect placement calculations in the real-time environment or in plans.

When calculating workload placement, Turbonomic respects cluster boundaries, networks, and provisioned data stores. In addition, the configuration of your environment can specify logical boundaries, and you can create even more boundaries within Turbonomic. These boundaries impose segments on the market that Turbonomic uses to model your application infrastructure.

In finance, a market segment divides the market according to the criteria different groups of people use when they buy or sell goods and services. Likewise in the Turbonomic market, a workload placement segment uses criteria to focus the buying and selling of resources within specific groups of entities. This gives you finer control over how Turbonomic calculates placements.

## Imported Placement Policies

Your on-prem targets can include placement policies of their own. Turbonomic imports these placement policies, and considers them to be constraints on placement. You cannot disable these imported policies for real-time analysis, but you can disable them for plans.

Turbonomic imports the following:

- vCenter Server DRS Rules  
See [Other Information Imported from vCenter](#)
- Virtual Machine Manager Availability Sets  
See [Virtual Machine Manager](#)
- Flexera One License Specifications  
See [Flexera](#)

To view imported placement policies, navigate to **Settings > Policies**, and then click **Imported Placement Policies**.

## Creating Placement Policies

A placement policy is a set of rules that Turbonomic must satisfy when generating placement recommendations for on-prem and containerized workloads.

With these policies, Turbonomic can recommend placement actions that comply with your business rules. For example, a placement policy can constrain VMs to specific hosts. If a VM needs to move to a different host due to resource congestion, Turbonomic will generate an action to move the VM to one of the hosts defined in the policy.

Turbonomic discovers placement policies from your targets, and allows you to create your own policies.

You can create the following placement policies:

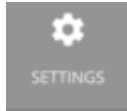
- **Place** — Determine which entities use specific providers  
For example, the VMs in a consumer group can only run on a host that is in the provider group. You can limit the number of consumers that can run on a single provider — for hosts in the provider group, only 2 instances of VMs in the consumer group can run on the same host. Or no more than the specified number of VMs can use the same storage device.
- **Don't Place** — Consumers must never run on specific providers  
For example, the VMs in a consumer group can never run on a host that is in the provider group. You can use such a segment to reserve specialized hardware for certain workloads.
- **Merge** — Merge clusters into a single provider group

For example, you can merge three host clusters in a single provider group. This enables Turbonomic to move workload from a host in one of the clusters to a host in any of the merged clusters to increase efficiency in your environment.

- **License** — Set up hosts to provide licenses for VMs

For VMs that require paid licenses, you can create placement policies that set up certain hosts to be the VMs' preferred license providers. Turbonomic can then recommend consolidating VMs or reconfiguring hosts in response to changing demand for licenses.

1. Navigate to the Settings Page.



Click to navigate to the Settings Page. From there, you can perform a variety of Turbonomic configuration tasks.

2. Choose Policies.



Click to navigate to the Policy Management Page.

This page lists all the policies that you currently have configured for Turbonomic.

3. Create a new Placement policy.

**NEW POLICY**

**Create a new policy** [X]

POLICY NAME  
[Text Input]

TYPE  
PLACE | DON'T PLACE | MERGE | LICENSE

PLACE  
Choose consumer type.. [Dropdown] [SELECT GROUP]

ON  
Choose provider type... [Dropdown] [SELECT GROUP]

Limit workload entities to placement group

Limit the maximum number of workload entities per placement entity to

SAVE POLICY

First, select the type of Placement policy to create, then specify the settings:

- Give the policy a name
- Choose the policy type and make the settings
- Save the policy when you're done

4. Create a **Place** policy.

POLICY NAME

TYPE

**PLACE** DON'T PLACE MERGE LICENSE

PLACE

Choose consumer type..

ON

Choose provider type...

Limit workload entities to placement group

Limit the maximum number of workload entities per placement entity to

These policies control where workload can be placed. For example, you can specify that a VM will only be placed on a host that is a member of a specific cluster. Or you could specify that any applications in a specific group can only be placed on a datastore that is a member of a specific group.

- **Specify the consumer group** — The group or cluster of entities that will be placed on the identified providers
- **Specify the provider group** — The group or cluster of entities that will provide resources to the consumers
- **Limit workload entities to placement group** — Set the policy to only place consumer entities on members of the provider group
- **Limit the maximum number of workload entities per placement entity to** — Limit how many instances of the consumer entities can be placed on a single provider

5. Create a **Don't Place** policy.

POLICY NAME

TYPE

PLACE    **DON'T PLACE**    MERGE    LICENSE

DON'T PLACE

Choose consumer type.. ▾    SELECT GROUP

ON

Choose provider type... ▾    SELECT GROUP

These policies identify groups or clusters that will never host the consumer entities. For example, you can specify that a VM will never be placed on a host that is a member of a specific cluster. Or you can specify that a set of non-critical applications will never be placed on specialized hardware, as a way to ensure availability for critical applications.

- **Specify the consumer group** — The group or cluster of entities that will be excluded from the identified providers
- **Specify the provider group** — The group or cluster of entities that will not provide resources to the consumers

6. Create a **Merge** policy.

POLICY NAME

TYPE

PLACE    DON'T PLACE    **MERGE**    LICENSE

MERGE

Choose consumer type.. ▾    SELECT CLUSTERS

You can create placement policies that merge multiple clusters into a single logical group for the purpose of workload placement.

For example, your environment might divide hosts into clusters according to hardware vendor, or by some other criteria. Workload placement typically does not cross such cluster boundaries. However, there might be no technical reason to apply these boundaries to workload placement. By creating a larger pool of provider resources, Turbonomic has even more opportunities to increase efficiency in your environment.

For merge policies, keep the following considerations in mind:

- For most policies that merge host and storage clusters, the clusters you place in the Merge segment must be members of the same datacenter.
- For vCenter environments, you can create placement policies that merge data centers to support cross-vCenter moves. In this case, where a data center corresponds to a given vCenter target, the merged clusters can be in different data centers. In this case you must create two merge policies; one to merge the affected data centers, and another to merge the specific clusters.

Also note that the clusters you merge must use the same network names on their respective datacenters.

To create a Merge policy, choose the type of entity to merge, and then select the groups you will merge.

#### 7. Create a **License** policy.

The screenshot shows a configuration form for a License policy. It includes a text input field for the policy name, a type selector with 'LICENSE' selected, and two sections for selecting license consumers and providers, each with a dropdown menu and a 'SELECT GROUP' button.

POLICY NAME

TYPE  
PLACE DON'T PLACE MERGE **LICENSE**

LICENSE  
Choose consumer type..

ON  
Choose provider type...

Assume you have purchased a number of licenses for a database – you pay for the right to run that database on a certain number of hosts. You can create a license policy to identify the hosts that provide the license, and the VMs that can consume that license.

After you create the policy, Turbonomic can recommend the following actions in response to changing demand for licenses:

- When demand is low, Turbonomic recommends consolidating VMs on as few license-providing hosts as possible to reduce your license costs. To consolidate, you move VMs to another host and then reconfigure the original hosts to remove their licenses. Note that Turbonomic will *not* recommend suspending these hosts. Since they remain active, they can be reconfigured to become providers when demand starts to exceed capacity.

For example, if you have Host\_01 providing a license to VM\_01 and Host\_02 providing a license to VM\_02, you will see two recommendations – move VM\_02 to Host\_01 and then remove the license in Host\_01. You will not see a recommendation to suspend Host\_01.

- When demand exceeds capacity, and there are hosts in the policy that currently do not provide licenses, Turbonomic recommends reconfiguring those hosts to become providers and then moving VMs to those hosts. If all hosts are currently providing licenses, Turbonomic recommends adding licenses to the hosts to meet demand.

These actions are more efficient than provisioning new hosts.

To create a License policy:

- Specify the license consumers (VMs).
- Specify the license providers (hosts).

In addition to creating a license policy, you must also create host *automation* policies to allow Turbonomic to recommend reconfigure actions on hosts. In the automation policies, add the license-providing hosts and then enable the *Reconfigure* action.

#### 8. When you have made all your settings, be sure to save the Policy.

## Automation Policies

An automation policy is a set of rules that Turbonomic must satisfy when executing non-parking actions on public cloud or on-prem workloads, or changing settings that affect analysis and action generation.

Automation policies include the following settings:

- Action Generation

This setting specifies whether a specific action will be generated.

- Action Acceptance

If a specific action will be generated, this setting specifies the degree of automation for the action. For more information, see [Action Acceptance Modes](#).

- Action Orchestration and Workflows

These settings specify whether Turbonomic or a workflow orchestrator will execute the action. For more information, see [Action Orchestration](#).

- Constraints and Other Settings

These settings affect the Turbonomic analysis of the state of your environment. These include operational, utilization, and scaling constraints.

The settings you can make are different according to the type of entity this policy will affect. Each setting you add to the policy takes precedence over the default value for that setting.

## Default and User-defined Automation Policies

Turbonomic ships with default automation policies that we believe will give you the best results from our analysis. For certain entities in your environment, you can create automation policies as a way to override the defaults.

For example, **Enforce Non Disruptive Mode** is turned off in the default automation policy for on-prem VMs. In most cases, you might want to turn on the setting, and only turn it off for select VMs. In that case, you would turn it on in the default automation policy for VMs, and then create policies for those groups of VMs for which you want to turn it off.

The default and user-defined automation policies take effect in relation to each other. A default policy has a global effect, while a user-defined policy overrides the default policy for the entities within the indicated scope. You should keep the following points in mind:

- User-defined policies override a subset of settings.

A user-defined policy can override a subset of settings for the entity type. For the remainder, Turbonomic will use the default policy settings on the indicated scope.

- When an entity applies conflicting user-defined policies, Turbonomic applies the following tie breakers:

- A scheduled policy always takes precedence over a non-scheduled policy, even if the non-scheduled policy is more conservative.
- Among scheduled policies with *identical* schedules, the most conservative setting wins.
- Among non-scheduled policies, the most conservative setting wins.

For example, a VM currently belongs to four groups with different policy settings.

- Group A policy: Resize VM in *Manual* mode every Saturday.
- Group B policy: Resize VM in *Automatic* mode every Saturday.
- Group C policy: Resize VM in *Manual* mode (no schedule).
- Group D policy: Resize VM in *Recommend* mode (no schedule).

Results:

- On a Saturday, Groups A and B policies take precedence over Groups C and D policies. The VM ultimately applies the Group A setting because it is more conservative.
- On all the other days, only Groups C and D policies are active. The VM applies the Group D setting because it is more conservative.

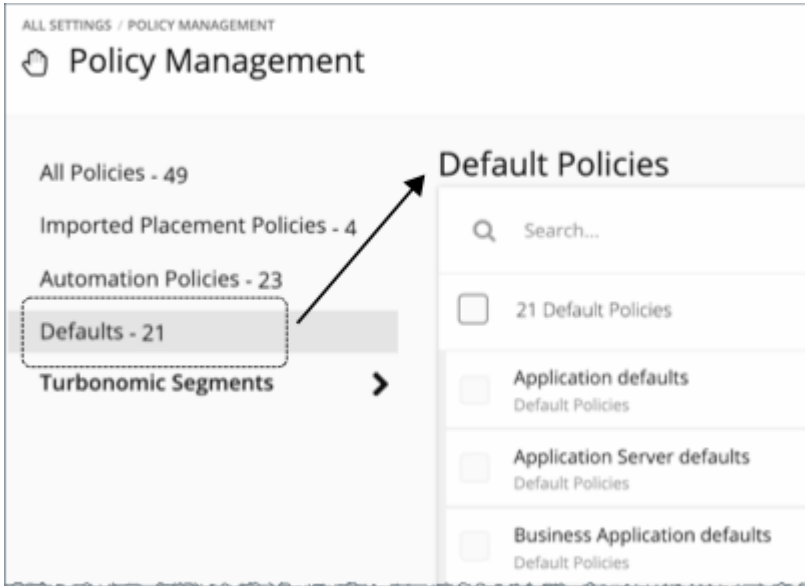
- User-defined policies always take precedence over default policies.

Even if the default policy has a more conservative setting, the setting in the user-defined policy wins for entities in that scope.

- For a global effect, *always* use default policies.

## Default Automation Policies

To view the default automation policies, navigate to **Settings > Policies**, and then click **Defaults**.



The page displays a list of all the default policies, by entity type. You can click the entity type to view or change the default settings.

## Global Default Policy

Use these settings to modify Turbonomic analysis globally for any scope of your environment. These defaults affect both default and user-defined automation policies.

## ACTION AUTOMATION

### Disable All Actions

| Attribute           | Default Setting |
|---------------------|-----------------|
| Disable All Actions | OFF             |

When this is ON, Turbonomic does not generate any actions for your environment. For example, assume you have configured a number of policies that automate actions, but you want to stop making changes to the entire environment for a period of time. Turn this ON to stop all execution with a single setting.

## OPERATIONAL CONSTRAINTS

### VM Growth Observation Period

| Attribute                    | Default Value |
|------------------------------|---------------|
| VM Growth Observation Period | 1 month       |

Use this setting to specify how much historical data the Turbonomic analysis will use to calculate time to exhaustion of your cluster resources.

Turbonomic runs nightly plans to calculate headroom for the clusters in your on-prem environment. To review your cluster headroom in dashboards, set the view scope to a cluster. With that scope, the view includes charts to show headroom for that cluster, as well as time to exhaustion of the cluster resources.

To calculate cluster growth trends, analysis uses historical data for the given clusters. With **VM Growth Observation Period**, you can specify how much historical data the headroom analysis will use to calculate time to exhaustion of your cluster resources. For example, if cluster usage is growing slowly, then you can set the observation to a period that is long enough to capture that rate of growth.

If the historical database does not include at least two entries in the monthly data for the cluster, then analysis uses daily historical data.

## Allow Unlimited Host Provisioning

| Attribute                         | Default Setting |
|-----------------------------------|-----------------|
| Allow Unlimited Host Provisioning | OFF             |

By default, Turbonomic allows overprovisioning hosts up to 10 times their memory capacity, and up to 30 times their CPU capacity. When this setting is ON, Turbonomic removes these overprovisioning limits to allow VM placements on already overprovisioned hosts.

This setting does not stop Turbonomic from recommending actions to provision new hosts in clusters.

## Enable Analysis of On-prem Volumes

| Attribute                          | Default Setting |
|------------------------------------|-----------------|
| Enable Analysis of On-prem Volumes | OFF             |

On-prem volumes represent VM Disks discovered by hypervisor targets. A VM will have one volume for each configured disk and another volume (representing the configuration) that always moves with Disk 1.

- **OFF** (default)

Turbonomic analyzes volume resources as part of VM analysis. In the real-time market and on-prem plans, any action to move VM storage ensures that volumes stay together on the underlying datastore. A [Migrate to Cloud plan](#) will recommend storage per datastore to hold all the VM Disks currently on the datastore.

For example, assume a VM with three disks. Disks 1 and 3 are on Datastore A, while Disk 2 is on Datastore B.

- During a storage migration, VM Disk volumes 1 and 3 will stay on the same datastore.
- A Migrate to Cloud plan will recommend a storage disk for VM Disk volumes 1 and 3, and another storage disk for VM Disk volume 2.

- **ON**

Turbonomic analyzes resources on each volume independently. In the real-time market and on-prem plans, any action to move VM storage migrates volumes to the most optimal datastore. A Migrate to Cloud plan will recommend storage for each volume.

For example, assume a VM with three disks. Disks 1 and 3 are on Datastore A, while Disk 2 is on Datastore B.

- During a storage migration, VM Disk volumes 1, 2, and 3 can migrate to different datastores.
- A Migrate to Cloud plan will recommend three separate storage disks for VM Disk volumes 1, 2, and 3.



### **Important:**

When you turn on this setting, your Turbonomic instance will start to use more memory and storage to perform its analysis. For example an environment with 10,000 VMs and an average of three disks per VM represents a three-fold increase in entities that require analysis. Currently, instances that monitor more than 50,000 VMs will experience a significant drop in performance. For this reason, this setting is turned off by default.

Before turning on this setting, review your [VM automation policies](#) and verify that Storage Move actions are in *Recommend* or *Manual* mode. In addition, review your [storage placement policies](#) to ensure that individual VM volumes can be placed on the expected storage.

## **Imported Automation Policies**

As Turbonomic discovers your environment, it can find configurations that set up scopes that need specific policies. For example:

- HA Configurations

For vCenter Server environments, Turbonomic discovers HA cluster settings and translates them into CPU and memory utilization constraints. The discovery creates a group of type *folder* for each HA cluster, and creates a policy that sets the appropriate CPU and memory constraints to that policy.

- Availability Sets

In public cloud environments, Turbonomic discovers groups of VMs that should keep all their VMs on the same template. In the Automation Policies list, these appear with the prefix `AvailabilitySet::` on the policy names. You can enable Consistent Resizing for the VMs in each group so Turbonomic can resize them to the same size.

To view imported automation policies, navigate to **Settings > Policies**, and then click **Imported Automation Policies**.

## **Creating Automation Policies**

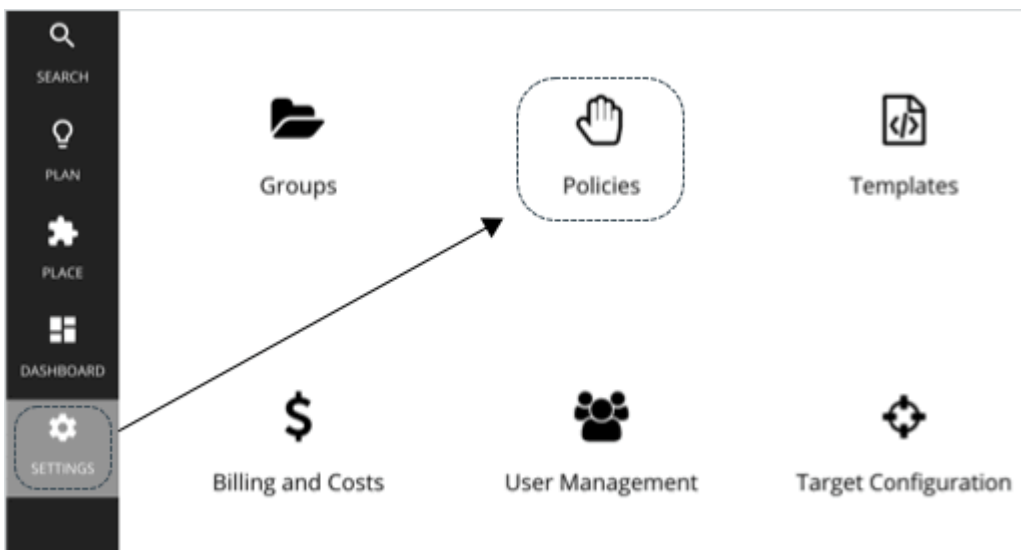
To override the default automation policies, you can create your own policies. These policies specify the settings you want to change for certain entities in your environment. You can assign a schedule to your policy to set up maintenance windows or other scheduled actions in your environment.

Below are some use cases for creating automation policies.

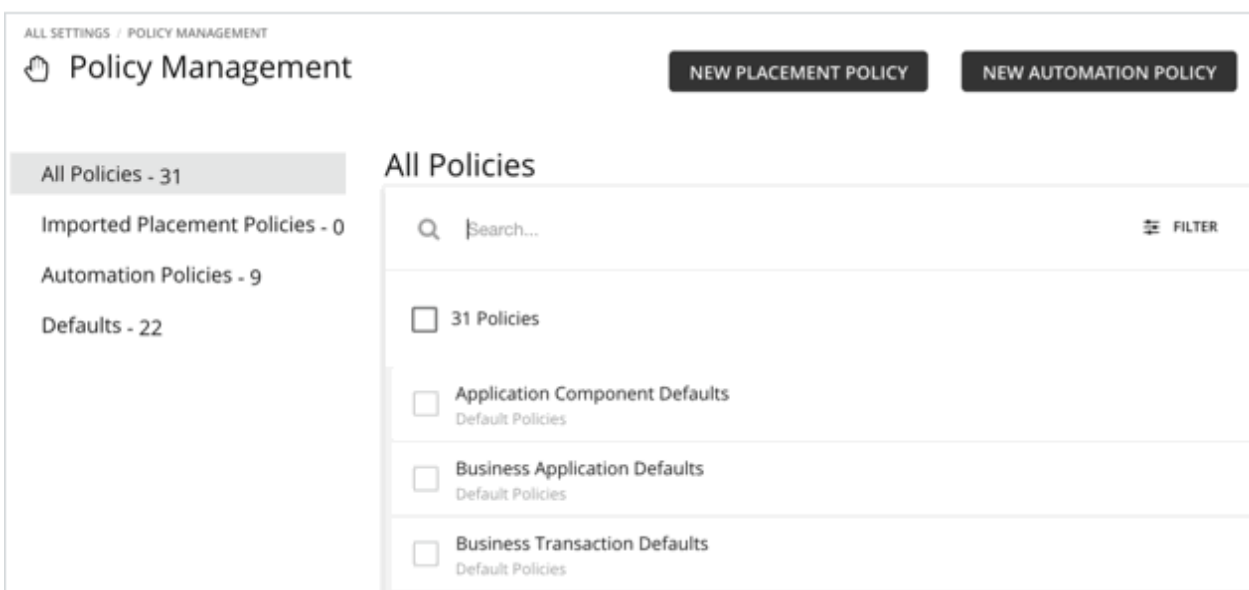
- Turbonomic uses a number of settings to guide its analysis of the entities in your environment. The default settings might be fine in most cases, but you might want different analysis for some groups of entities.
- Assume you want to automate scaling and placement actions for the VMs in your environment. It is common to take a cautious approach, and start by automating clusters that are not critical or in production. You can scope the policy to those clusters, and set the action acceptance mode to Automatic for different actions on those VMs (see [Action Acceptance Modes](#)).
- If actions require change approval, or integrations into DevOps pipelines to perform tasks before, instead of, or after action execution, you should scope those entities to a group and apply an automation policy.

### **1. Entry Point**

Navigate to the Settings Page and then choose **Policies**.



This opens the Policy Management Page, which lists all the currently available policies.



Click **NEW AUTOMATION POLICY** and then select the policy type (such as Virtual Machine).

This sets the type of entity that your policy will affect. Note that Turbonomic supports different actions for different types of entities. For example, you cannot add VMem to a storage device. Setting policy type is the first step you take to focus on which actions you want to map to your workflows.

## 2. Policy Name

Name the policy.

< **Configure Virtual Machine Policy** ×

NAME  
VM\_Policy

+ SCOPE

+ POLICY SCHEDULE

+ AUTOMATION AND ORCHESTRATION

+ ACTION CONSTRAINTS

+ OPERATIONAL CONSTRAINTS

+ SCALING CONSTRAINTS

SAVE AND APPLY

### 3. Scope

The screenshot shows a 'Configure Virtual Machine Policy' window. At the top, there's a title bar with a back arrow and a close 'X' button. Below that is a 'NAME' field with the text 'VM\_Policy'. A dashed box highlights the 'SCOPE' section, which contains a tag 'ACM x' and a button 'ADD VIRTUAL MACHINE GROUPS'. Below the 'SCOPE' section are five expandable sections: 'POLICY SCHEDULE', 'AUTOMATION AND ORCHESTRATION', 'ACTION CONSTRAINTS', 'OPERATIONAL CONSTRAINTS', and 'SCALING CONSTRAINTS'. At the bottom of the dialog is a 'SAVE AND APPLY' button.

The scope determines which entities this policy will affect. Choose one or more groups, or create new groups and add them to the policy scope. These groups match the type of entity you have set for the policy.

In Turbonomic you can find nested groups (groups of groups). For example, the "By PM Cluster" group contains host clusters, and each host cluster is a group. Do not set the policy scope to a parent of nested groups. When setting up policies, be sure you set them to individual groups. If necessary, create a custom group for the settings you want to apply.

**Note:**

A single entity can be a member of multiple groups. This can result in a conflict of settings, where the same entity can have different policy settings. For conflicts among user-defined policy settings, the most conservative setting will take effect. For details, see [Default and User-defined Policies](#).

## 4. Policy Schedule

The screenshot shows the 'Configure Virtual Machine Policy' interface. The main window has a title bar with a back arrow and a close 'X' button. Below the title is a 'NAME' field containing 'VM\_Policy'. There are several expandable sections: '+ SCOPE', '- POLICY SCHEDULE', '+ AUTOMATION AND ORCHESTRATION', '+ ACTION CONSTRAINTS', '+ OPERATIONAL CONSTRAINTS', and '+ SCALING CONSTRAINTS'. A 'SAVE AND APPLY' button is at the bottom. An 'ATTACH SCHEDULE' link with a right-pointing arrow is highlighted, leading to a 'Select Schedule' fly-out menu. The fly-out menu has a search bar, a 'FILTER' button, and a 'NEW SCHEDULE' button. It lists 'All Schedules (21)'. Two schedules are visible: 'Ap' (Does not recur, EXPIRED) and 'Ap 2' (Daily, STARTS IN 8 HOURS). The 'Ap 2' schedule is selected with a checkmark. Below the list is a 'SCHEDULE DETAILS' section for 'Ap 2' with the following information:

- SCHEDULE DETAILS**
- SUMMARY**: Occurs every day effective Fri, Apr 3, 2020 starting at 3:30 AM and finishing same day at 5:30 AM.
- USED IN POLICIES**: 0
- NEXT OCCURRENCE**

| DATE                        | TIME          |
|-----------------------------|---------------|
| Saturday, December 12, 2020 | 3:30 AM (UTC) |
- ACCEPTED ACTIONS**: 0
- AWAITING ACCEPTANCE**: 0

A 'SET' button is at the bottom of the fly-out menu.

For use cases and information about how schedules affect policies, see [Policy Schedules](#).

The **Select Schedule** fly-out lists all the schedules that are currently defined for your instance of Turbonomic.

Expand a schedule entry to see its details. The details include a summary of the schedule definition, as well as:

- **Used in Policies**

The number of policies that use this schedule. Click the number to review the policies.

- **Next Occurrence**

When the schedule will next come into effect.

- **Accepted Actions**

How many scheduled actions have been accepted to be executed in the next schedule occurrence. Click the number for a list of these actions.

- **Awaiting Acceptance**

The number of Manual actions affected by this schedule that are in the Pending Actions list, and have not been accepted. Click the number for a list of these actions.

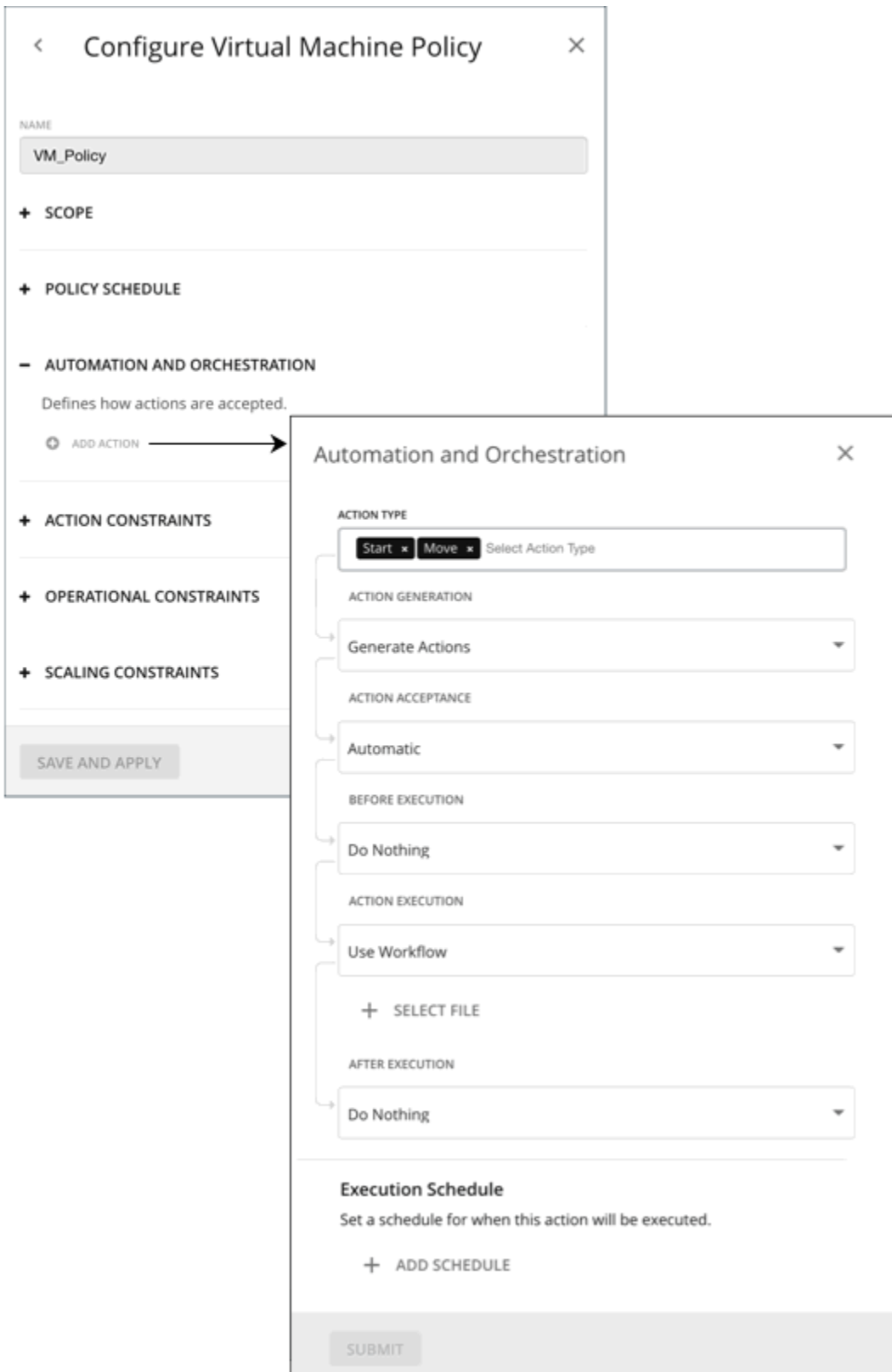
If none of the listed schedules is suitable for your policy (or if none exists), click **New Schedule**. For details, see [Managing Calendar Schedules](#).

**Note:**

When you configure a schedule window for a VM resize action, to ensure Turbonomic will execute the action during the scheduled time, you must turn off the **Enforce Non Disruptive Mode** setting for that scheduled policy. Even if you turn the setting off for the global policy, you still must turn the setting off for your scheduled policy. Otherwise Turbonomic will not execute the resize action.

## **5. Automation and Orchestration**

You can define automation and orchestration settings for different action types within the same policy. For example, for a group of VMs in a policy, you can automate all *Resize* actions, but require *Suspend* actions to go through an approval process via an Orchestrator (such as ServiceNow).



### 5.1. Action Type

See a list of actions that are viable for the policy, and then make your selections.

### 5.2. Action Generation and Acceptance

- Do not Generate Actions

Turbonomic never considers your selected actions in its calculations. For example, if you do not want to generate *Resize* actions for VMs in the policy, analysis will still drive toward the desired state, but will do so without considering resizes.

- **Generate Actions**

Turbonomic generates your selected actions to address or prevent problems. Choose from the following *Action Acceptance* modes to indicate how you would like the actions to execute:

- **Recommend** — Recommend the action so a user can execute it via the given hypervisor or by other means
- **Manual** — Recommend the action, and provide the option to execute that action through the Turbonomic user interface
- **Automatic** — Execute the action automatically

For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to execute all actions at once. Any action awaiting execution stays in queue. For example, if a VM has both vCPU and vMem resize actions, Turbonomic could resize vCPU first. After this resize completes, it waits five minutes before resizing vMem.

If you have an orchestrator target (such as ServiceNow), and that target includes an installation of the *Turbonomic Actions* application, you can send the action to the orchestrator. Choose from the following options:

- Generate Action then Send Record to Orchestrator
- Generate Action then Request Approval from Orchestrator

For more information, see [Action Orchestration](#).

### **5.3. Before Execution, Action Execution, and After Execution**

By default, generated actions execute without the need for orchestration. Turbonomic gives you the ability to set up orchestration to affect the execution of actions.

For more information, see [Action Orchestration](#).

### **5.4. Execution Schedule**

You can defer the execution of generated actions to a non-critical time window. For example, if a workload experiences memory bottlenecks during the week, you can defer the necessary resize to the weekend. Even if the workload has minimal utilization over the weekend, Turbonomic can recognize the need to resize, and will execute the action.

For more information, see [Action Execution Schedules](#).

## **6. Constraints and Other Settings**

Turbonomic collects metrics to drive the analysis that it uses when it calculates actions for your environment. It compares current utilization and demand against allocated capacities for resources, so it can recommend actions that keep your environment in optimal running condition.

Automation policies include constraints and other settings that you can make to adjust the analysis that Turbonomic performs. For example, you can set different levels of overprovisioning for host or VM resources, and Turbonomic will consider that as a factor when deciding on actions.

The settings you can make are different according to the type of entity this policy will affect. Each setting you add to the policy takes precedence over the default value for that setting.



Configure Virtual Machine Policy

NAME  
VM\_Policy

+ SCOPE

+ SCHEDULE

+ AUTOMATION AND ORCHESTRATION

- ACTION CONSTRAINTS  
+ ADD ACTION

- OPERATIONAL CONSTRAINTS  
+ ADD CONSTRAINT

- SCALING CONSTRAINTS  
+ ADD SCALING CONSTRAINT

SAVE AND APPLY

## Policy Schedules

You can set a schedule for an automation policy, which sets a window of time when the policy takes effect. For example, you can modify the operational or scaling Constraints for a given period of time. These settings affect Turbonomic analysis, and the actions that the platform generates. You can set up scheduled times when you want to change those settings.

Remember that for user-defined automation policies, it is possible that one entity can be in two different scopes. This means that the entity can be under the effect of two different policies. For this reason, user-defined policies keep the rule, *the most conservative setting wins*. However, a more aggressive user-defined policy takes precedence over the corresponding default automation policy.

You must consider these rules when you add schedules to policies. If the more conservative settings are in a default automation policy, then the scheduled change takes effect. However, if the more conservative settings are in another user-defined policy, then the conservative settings *win*, and the scheduled changes do not take effect.

For details, see [Default and User-defined Policies](#).

## Policy Schedule and Action Execution Schedule

A scheduled policy can include *actions*. When the policy is in effect, Turbonomic recommends or automatically executes those actions as they are generated. Some of those actions could be disruptive so you may want to defer their execution to a non-critical time window. In this case, you will need to set an *action execution schedule* within the scheduled policy. For example, you can set a policy that automatically resizes or starts VMs for your customer-facing apps for the entire month of December, in

anticipation of an increase in demand. Within this same policy, you can set the resize execution schedule to Monday, from midnight to 7:00 AM, when demand is expected to be minimal.

For more information, see [Action Execution Schedules](#).

## Action Execution Schedules

You can defer the execution of generated actions to a non-critical time window. For example, if mission-critical VMs experience memory bottlenecks during the week, you can defer the necessary memory resizes to the weekend. Even if the VMs have minimal utilization over the weekend, Turbonomic can recognize the need to resize, and will execute resize actions. For this particular example, you will need to:

1. Create a policy for the VMs.
2. Select *VMem Resize Up* from the list of actions and then set the action mode to either *Automatic* or *Manual*.

### Note:

Execution schedules have no effect on recommended actions. It is therefore not necessary to set up an execution schedule if all the actions in your policy will be in *Recommend* mode.

3. Set an Execution Schedule that starts on Saturday at 8:00 AM and lasts 48 hours.

## Execution of Scheduled Actions

Turbonomic posts an action at the time that the conditions warrant it, which means that you might see the action in the Pending Actions list even before the execution schedule takes effect. The action details show what schedule affects the given action, and shows the next occurrence of that schedule.

- Automatic

When the schedule takes effect, Turbonomic executes any pending automated actions.

- Manual

Before the execution schedule, the action details for manually executable actions show the action state as `PENDING ACCEPT`. If you accept the action (select it and click **Apply Selected**), then Turbonomic adds it to the queue of actions to be executed during the maintenance window. The action details show the action state as `AWAITING EXECUTION`. Turbonomic executes the actions when the schedule takes effect.

## Keeping Actions Valid Until the Scheduled Time

If you have scheduled action execution for a later time, then conditions could change enough that the action is no longer valid. If this happens, and the action remains invalid for 24 hours, then Turbonomic removes it from the list of pending actions. This action will not be executed.

Turbonomic includes scaling constraints that work to stabilize action decisions for VMs. The resulting actions are more likely to remain valid up until their scheduled window for execution. You can make these settings in the default or user-defined automation policies.

### Note:

When you configure an execution schedule for a resize action, to ensure Turbonomic will execute the action during the scheduled time, you must turn off the **Enforce Non Disruptive Mode** setting for the policy. Even if you turn the setting off for the global policy, you still must turn the setting off for your policy. Otherwise Turbonomic will not execute the resize action. For information about non disruptive mode, see [Non-disruptive Mode](#).

## Action Orchestration

Action Orchestration specifies whether Turbonomic will execute an action, or whether Turbonomic will pass the action request to an orchestrator or an action workflow to effect the change in your environment.

In this way, you can integrate supported orchestrators to execute of actions for specific scopes of entities in your environment.

## About Orchestrators

Action Orchestration targets assign workflows that execute multiple actions to make changes in your environment. Turbonomic discovers workflows that you have defined on the orchestrator. You can then set up an automation policy that maps workflows to actions. If the action acceptance mode is *Manual* or *Automatic*, then when Turbonomic recommends the action, it will direct the orchestrator to use the mapped workflow to execute it.

Turbonomic supports integration with ServiceNow. You can configure policies that log Turbonomic actions in your ServiceNow instance, and that submit actions for approval in ServiceNow workflows.

This section shows how to link orchestration workflows to automation policies. It assumes you have already configured an appropriate Orchestration target. It also assumes that you have configured workflows on that target in such a way that Turbonomic can discover the workflows and map them to automation policies. For information about Orchestration target requirements, see [Orchestrator Targets](#).

### Note:

For some orchestration workflows, it is necessary to schedule an action to execute only during a specific maintenance window. Turbonomic policies can include schedules to enable this use case. However, you must be sure that you do not set the schedule to the policy that declares the orchestration you want. Instead, you should use two policies for the same scope – one to set up the orchestration, and another to schedule the time window during which the action acceptance mode will be *Automatic* (to set up the maintenance window). For more information, see [Setting Policy Schedules](#).

## About Action Workflows

Action workflows can add custom processing to Turbonomic actions at different entry points. For example, you can create an action workflow that sends an email whenever Turbonomic recommends moving a VM, or you can create an action workflow that runs as a replacement for the action that Turbonomic would execute. For example:

- Action Script workflow

You can deploy action scripts on a remote machine and configure an Action Script target that communicates with this server. Turbonomic discovers the exposed scripts and displays them as options you can choose when you specify a workflow in your automation policy. For more information about Action Scripts, see [Deploying Action Scripts](#).

- Webhook workflow

To implement a webhook, you create a workflow that specifies parameters such as the HTTP URL, HTTP method, and payload template. You can then use this workflow in your automation policy to orchestrate how actions execute. For more information about webhooks, see [Creating Webhook Workflows](#).

## Specifying Action Orchestration

As you create a policy, you specify the entity type and the scope of entities the policy affects. You can also set modes for specific actions. For example, you can set a mode of *Manual* for the *Resize* action for a given scope of VMs.

1. Expand **Automation and Orchestration** and click **Add Action**. Then select the action type you want to orchestrate.

< Configure Virtual Machine Policy
×

NAME

VM\_Policy

**+ SCOPE**

---

**+ POLICY SCHEDULE**

---

**- AUTOMATION AND ORCHESTRATION**  
Defines how actions are accepted.

+ ADD ACTION →

---

**+ ACTION CONSTRAINTS**

---

**+ OPERATIONAL CONSTRAINTS**

---

**+ SCALING CONSTRAINTS**

---

SAVE AND APPLY

Automation and Orchestration
×

**ACTION TYPE**

Start Move Select Action Type

**ACTION GENERATION**

Generate Actions

**ACTION ACCEPTANCE**

Automatic

**BEFORE EXECUTION**

Do Nothing

**ACTION EXECUTION**

Use Workflow

+ SELECT FILE

**AFTER EXECUTION**

Do Nothing

---

**Execution Schedule**  
Set a schedule for when this action will be executed.

+ ADD SCHEDULE

SUBMIT

There is no orchestration for this action by default. The following table describes the supported orchestrations.

|                   | <b>Generate Actions</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <b>Generate Action then Send Record to Orchestrator</b>                          | <b>Generate Action then Request Approval from Orchestrator</b>                                                                                                                     |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description       | Generate actions as usual, but use a workflow to control action execution.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Send a record of the generated actions to the orchestrator (such as ServiceNow). | Defer the generated actions to your orchestration workflow for approval.<br><br>Turbonomic passes control for this action to your orchestration workflow as a Change Request (CR). |
| Prerequisites     | An Orchestration target                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | An Orchestration target                                                          | An Orchestration target                                                                                                                                                            |
| Action Acceptance | Choose from the following: <ul style="list-style-type: none"> <li>• Recommend – Recommend the action so a user can execute it via the given hypervisor or by other means</li> <li>• Manual – Recommend the action, and provide the option to execute that action through the Turbonomic user interface</li> <li>• Automatic – Execute the action automatically</li> </ul> For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to execute all actions at once. Any action awaiting execution stays in queue. For example, if a VM has both vCPU and vMem resize actions, Turbonomic could resize vCPU first. After this resize completes, it waits five minutes before resizing vMem. |                                                                                  | Action acceptance automatically changes to "External Approval." If the action is approved, the action executes using the default Action Acceptance mode.                           |
| Before Execution  | The default is Do Nothing.<br><br>Select <b>Use Workflow</b> to trigger an external orchestration workflow such as a webhook to complete pre-execution tasks. Failure of the workflow indicates an action failure. For example, you can post an email notification to your team that an action has been generated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                  |                                                                                                                                                                                    |
| Action Execution  | The default is Native. Turbonomic executes the action with its default action processing.<br><br>Select <b>Use Workflow</b> to trigger an external orchestration workflow to execute the action. Failure of the workflow indicates an action failure.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                  |                                                                                                                                                                                    |

|                    | <b>Generate Actions</b>                                                                                                                                                                                                                | <b>Generate Action then Send Record to Orchestrator</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | <b>Generate Action then Request Approval from Orchestrator</b>                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Execution Schedule | <p>There is no execution schedule by default. Turbonomic executes the action immediately.</p> <p>If the policy includes a schedule, Turbonomic executes the action at the scheduled time.</p>                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | <p>There is no execution schedule by default. Turbonomic executes the action immediately.</p> <p>If the policy includes a schedule, Turbonomic executes the action at the scheduled time.</p> <p><b>Note:</b></p> <p>Turbonomic discovers and enforces execution schedules defined in orchestration approval workflows. To avoid potential issues with schedules, set the execution schedule either in the orchestrator (such as ServiceNow) or Turbonomic.</p> |
| After Execution    | <p>The default is Do Nothing.</p> <p>Select <b>Finalize action workflow</b> to trigger an external orchestration workflow such as a webhook to complete post-execution tasks. Failure of the workflow indicates an action failure.</p> | <p>The default is Do Nothing.</p> <p>Other options include:</p> <ul style="list-style-type: none"> <li>• <b>Notify action workflow</b></li> </ul> <p>Trigger an external orchestration workflow such as a webhook. For example, you can post a message to your collaboration platform to let your team know that an action has executed.</p> <p>Action status is not affected by the status of the notification workflow.</p> <ul style="list-style-type: none"> <li>• <b>Finalize action workflow</b></li> </ul> <p>Trigger an external orchestration workflow such as a webhook to complete post-execution tasks. Failure of the workflow indicates an action failure.</p> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

2. When you have made all your settings, be sure to save the action policy.

## Deploying Action Scripts

Action Scripts provide an interface that can add custom processing to Turbonomic actions.

Action scripts execute on a remote server (a VM or a container) that you have configured as a Turbonomic target. That server includes a manifest file that identifies the scripts you have deployed, as well the entities and actions they can respond to. Turbonomic discovers these scripts via the manifest and presents them as orchestration options for actions in automation policies.

For example, assume you have defined a script with:

- name: MyVmMoveAction
- entityType: VIRTUAL\_MACHINE

- `actionType: MOVE`

Following this example, you can use the API to add orchestration to a policy for move actions on VMs. Because you have defined a script for that action, you can specify Action Script as the orchestration type, and you can choose the `MyVmMoveAction` script as the orchestration workflow to perform.

## **Setting Up the Action Script Server**

Turbonomic uses remote servers to execute action scripts. Managing the processes remotely means that you do not install custom code on the Turbonomic server, which eliminates associated security risks there. However, you are responsible for maintaining the security of your Action Script Server to ensure the integrity of your custom code. To accomplish this, the configuration of the remote server must meet certain requirements.

## **Resource Requirements for the Server**

The remote server can be a VM or a container. The capacity you configure for the server depends entirely on the processes you intend to run on it. Turbonomic does not impose any special resource requirements on the server.

## **Configuring Command Execution**

To support execution of your scripts, you must install any software that is necessary to run the scripts. This includes libraries, language processors, or other processes that your scripts will invoke.

Turbonomic invokes the scripts as commands on the server. The server must run an SSH service that you have configured to support command execution and SFTP operations. At this time, IBM has tested action scripts with the OpenSSH `sshd` daemon.

The standard port for SSH is 22. You can configure a different port, and provide that for admins who configure the server as an Action Script target.

An action script can invoke any process you have deployed on the remote server. You do not have to run scripts; however, you must be able to invoke the processes from the command line. The script manifest gives Turbonomic the details it needs to build the command line invocation of each script.

## **Configuring the Action Script User Account**

To execute the scripts on your server, Turbonomic logs on via a user account that is authorized to execute the scripts from the command line. You provide the user credentials when you configure the Action Script target. To support this interaction, the user account must meet the following requirements:

- Public key

The user must have a public key in the `.ssh/authorized_keys` file. When you configure the Action Script target, you provide this as the Private Token for the target.

- Security for the `.ssh` directory

The Action Script User should be the only user with authorized access. You should set file permissions to 600.

- Supported shells

The Action Script User shell can be either the Bourne shell (usually at `/bin/sh`) or the Bourne-Again shell (usually at `/bin/bash`). Turbonomic passes parameters as it invokes your scripts. At this time it only supports script execution through these shells.

## **Handling Action Script Timeouts**

Turbonomic limits script execution to 30 minutes. If a script exceeds this limit, Turbonomic sends a `SIGTERM` to terminate the execution of the process.

Note that Turbonomic does not make any other attempt to terminate a process. For example you could implement the script so it traps the SIGTERM and continues to run. The process should terminate at the soonest safe opportunity. However, if the process does not terminate, then you must implement some way to terminate it outside of Turbonomic. A runaway process continues to use its execution thread, which can block other processes (action scripts or primary processes) if there are no more threads in the pool.

## ***Creating Action Scripts***

An action script can be any executable that a user can invoke from a command line. You can save these executable files anywhere on the server – The Manifest indicates the path to the file (see [Deploying the Action Script Manifest](#)). The Action Script user that you have configured for the script server must have access to your script files, with read and execution privileges.

To execute a script, Turbonomic builds the appropriate SSH command from the manifest information it has discovered. It grants a timeout limit of 30 minutes by default, or the manifest entry can declare a different limit. If the execution exceeds the limit, Turbonomic sends a SIGTERM to terminate the process.

## **Passing Information to the Action Script**

Turbonomic uses two techniques to pass information about an action to the associated action script:

- Pass general information via environment variables
- Pass full action data via `stdin`

To pass general information into the script, Turbonomic sets environment variables on the Action Script Server. You can reference these environment variables in your scripts. For example, assume you want to send an email that includes the name of the VM that is an action target. You can get that name via the `VMT_TARGET_NAME` environment variable.

The following list shows the environment variables that Turbonomic can set when it executes a script. Note that not all of these variables apply for every action. For example, an action to scale VMEM does not include providers, so the action does not include values for the `VMT_CURRENT_INTERNAL`, `VMT_CURRENT_NAME`, `VMT_NEW_INTERNAL`, or `VMT_NEW_NAME` variables. If a given variable does not apply, Turbonomic sets it to an empty string.

- `VMT_ACTION_INTERNAL`  
The UUID for the proposed action. You can use this to access the action via the REST API. For example, your script could accept or cancel the action according to its own criteria.
- `VMT_ACTION_NAME`  
The name of the action.
- `VMT_CURRENT_INTERNAL`  
The internal name for the current provider.
- `VMT_CURRENT_NAME`  
The display name for the current provider.
- `VMT_NEW_INTERNAL`  
The internal name for the new provider.
- `VMT_NEW_NAME`  
The display name for the new provider.
- `VMT_TARGET_INTERNAL`  
The internal name of the entity this action will affect. You can use this to access the target entity via the REST API. For example, you can get historical statistics or you can change settings for the entity.
- `VMT_TARGET_NAME`  
The display name of the entity this action will affect.



- VMT\_TARGET\_UUID

The UUID of the entity this action will affect.

For some scripts, you might need a complete description of the associated action. For example, assume you want to analyze the utilization metrics for a given resource. The environment variables for passing general information do not include this information.

When it invokes an action script, Turbonomic passes the complete data for the associated action via `stdin`. Your script can load this into a variable to access the specific data it needs. For example, the following loads `stdin` into `myActionData`:

```
myActionData=$(cat -)
```

`stdin` contains a JSON string that represents of the full data associated with this action. For example, the `myActionData` variable could contain a string similar to:

```
{"actionType":"RIGHT_SIZE","actionItem":
 [{"actionType":"RIGHT_SIZE","uuid":"143688943343760","targetS
 E":{"entityType":"VIRTUAL_MACHINE","id":"4200fcdb-eafe-2a4a-abf5-a7ad2b00555c"...
```

## Deploying the Action Script Manifest

The Action Script Manifest identifies the scripts that you want to expose to Turbonomic. You provide the location of the manifest as part of the Action Script Target configuration – After Turbonomic validates the target, it then discovers these scripts and presents them in the Orchestration Policy user interface.

## Creating the Scripts Manifest File

The Scripts Manifest is a file that declares an array of Script Objects for each script you want to expose. You can create the manifest as either a JSON or a YAML file.

For example, following are two examples of the same manifest – One in YAML and the other in JSON. Notice that in either case, the manifest is an array of two Script objects:

- YAML Manifest:

```
scripts:
 - name: MyVmMovePrep
 description: Execute this script in preperation to a VM Move
 scriptPath: vmScripts/movePrep.sh
 entityType: VIRTUAL_MACHINE
 actionType: MOVE
 actionPhase: PRE
 - name: MyVmSuspendReplace
 description: Execute this instead of a VM Suspend action
 scriptPath: vmScripts/suspendReplace.sh
 entityType: VIRTUAL_MACHINE
 actionType: SUSPEND
 actionPhase: REPLACE
```

- JSON Manifest:

```
{
 "scripts": [
 {
 "name": "MyVmMovePrep",
 "description": "Execute this script in preperation to a VM Move",
 "scriptPath": "vmScripts/movePrep.sh",
 "entityType": "VIRTUAL_MACHINE",
 "actionType": "MOVE",
 "actionPhase": "PRE"
 },
 {
 "name": "MyVmSuspendReplace",
 "description": "Execute this instead of a VM Suspend action",
 "scriptPath": "vmScripts/suspendReplace.sh",
```

```

 "entityType": "VIRTUAL_MACHINE",
 "actionType": "SUSPEND",
 "actionPhase": "REPLACE"
 }
]
}

```

You can save the Scripts Manifest file to any location on your server, so long as the Scripts User has access to that location, and has read and execute privileges. You will provide this location as the **Script Path**, which the Turbonomic administrator will give as part of the Action Script target configuration.

Note that the filename extension for the manifest must match the file format (either YAML or JSON). For example, you should name the file either `MyManifest.yaml` or `MyManifest.json`, respectively.

## Declaring Script Objects

Each script object in the manifest can contain the following fields:

- **name**

Required – The name for this action script. After Turbonomic discovers your scripts, it displays this name as a Orchestration Workflow choice in the user interface for creating orchestration policies.

- **description**

Optional – A description of the script. The Turbonomic user interface does not display this description.

- **scriptPath**

Required – The path to the executable for this entry. You can give an absolute path, or a path that is relative to the location of the Scripts Manifest. The Action Script User that you set up for the Action Script server must have read and execute privileges for the executable file.

- **entityType**

Required – The type of entity this script responds to. Can be one of:

- Switch
- VIRTUAL\_DATACENTER
- STORAGE
- DATABASE\_SERVER
- WEB\_SERVER
- VIRTUAL\_MACHINE
- DISK\_ARRAY
- DATA\_CENTER
- PHYSICAL\_MACHINE
- CHASSIS
- BUSINESS\_USER
- STORAGE\_CONTROLLER
- IO\_MODULE
- APPLICATION\_SERVER
- APPLICATION
- CONTAINER
- CONTAINER\_POD
- LOGICAL\_POOL
- STORAGE\_VOLUME
- DATABASE

- VIEW\_POD
- DESKTOP\_POOL

To configure the same script to respond to actions on different entity types, declare separate entries for that script, one for each entity type.

- **actionType**

Required – The type of action this script responds to. Note that different entity types can support different actions. Can be one of:

- START
- MOVE
- SCALE

Resize on cloud - move workload from one cloud template or tier to another.

- SUSPEND
- PROVISION
- RECONFIGURE
- RESIZE
- DELETE
- RIGHT\_SIZE
- ACTIVATE
- DEACTIVATE
- BUY\_RI

- **actionPhase**

Required – Where in the life cycle of an action that you want your script to execute.

Can be one of:

- PRE

For an action that has been accepted, or an AUTOMATED action before it executes, this state is a preparation phase where your script can execute just before the action itself executes.

Run your script to set up conditions just before the action executes.

- REPLACE

For action execution, your script executes *in stead of* the execution that Turbonomic would perform.

Run your script as a replacement for the Turbonomic action.

- POST

The action has completed execution, either in a SUCCEEDED, FAILING, or FAILED state.

FAILING means that the status was checked after the action execution fails, but before the POST script has finished execution.

Run your script after the action has completed execution.

- **timeLimitSeconds**

Optional – How long to run the action before assuming a timeout. When execution exceeds this limit, Turbonomic sends a SIGTERM to terminate the execution of the process.

If you do not provide a value, Turbonomic assumes a limit of 30 minutes (1800 seconds).

## Webhooks

You can configure automation policies in Turbonomic to send data via webhooks to external web servers. A webhook is an automated message that Turbonomic can use to send data to external applications. Things you can do with webhooks include:

- Send notifications to collaboration platforms such as Slack
- Integrate Turbonomic with workflow management systems
- Override Turbonomic actions with your own logic

The webhook implementation supports HTTP messaging. In addition, to implement a webhook you create a workflow via the Turbonomic API.

To set up a webhook, you will:

- Identify the application to receive the webhook

Possible applications can include collaboration platforms such as Slack, orchestration platforms such as ServiceNow, cloud provider APIs, or you can create a custom application that responds to HTTP methods.

- Create a webhook workflow in your Turbonomic instance

For this version of Turbonomic you define webhook workflows via the API.

A webhook definition can include:

- The URL to the application to which you are sending the webhook
- An HTTP method
- A template for the webhook payload
- Authentication credentials to access the application

For information about creating a webhook workflow, see [Creating Webhook Workflows](#).

- Create an Automation Policy that uses the webhook

Automation Policies include orchestration settings where you can choose to execute a webhook for given actions. Turbonomic can execute a webhook when it creates an action, before it executes the action, instead of executing the action, and after it executes the action.

For information about creating policies that use webhook workflows for orchestration, see [Action Orchestration](#).

After you set up a webhook, when Turbonomic generates or executes the action you identified in the policy, it sends a message to the url that you specified in the webhook.

### ***Creating Webhook Workflows***

To implement a webhook, you create a workflow that specifies parameters such as the HTTP URL, HTTP method, and payload template. You can then use this workflow in your automation policy to orchestrate how actions execute.

To create a workflow, use the API to POST a Workflow object to Turbonomic instance. For example, the following `curl` commands get authorization to access a Turbonomic server, and then add a simple webhook workflow to that server:

- Authenticate on the server

This command requests authentication credentials and stores them in a variable you can set to a cookie in subsequent `curl` headers, where:

- `<T8c_IP_ADDRESS>` is the address of the Turbonomic server
- `<ADMIN_ACCOUNT_NAME>` is the name of an account with admin privileges

- <ADMIN\_PWD> is the admin account password

```
JSESSIONID=$(curl \
 --silent \
 --cookie-jar - \
 --insecure \
 https://<T8c_IP_ADDRESS>/vmturbo/rest/login \
 --data "username=<ADMIN_ACCOUNT_NAME>&password=<ADMIN_PWD>" \
 | awk '/JSESSIONID/{print $7}')
```

- Create the workflow

This command creates the workflow on the server, where:

- <T8c\_IP\_ADDRESS> is the address of the Turbonomic server
- <WEBHOOK\_ADDRESS> is the address of the webhook server

```
curl \
 "https://<T8c_IP_ADDRESS>/api/v3/workflows" \
 --insecure \
 --compressed \
 --header 'Accept: application/json' \
 --header 'Content-Type: application/json' \
 --header "cookie: JSESSIONID=$JSESSIONID" \
 --request POST \
 --data '
{
 "displayName": "My_WebHook",
 "className": "Workflow",
 "description": "First webhook attempt.",
 "discoveredBy":
 {
 "readonly": false
 },
 "type": "WEBHOOK",
 "typeSpecificDetails": {
 "url": "http://<WEBHOOK_ADDRESS>",
 "method": "POST",
 "template": "{ \"text\": \"My Webhook Template -- DATA: Action Details:
$action.details\" }",
 "type": "WebhookApiDTO"
 }
}
```

This is a simple webhook that sends its template to the indicated url. For a listing of the parameters you can set in the workflow, see [WebhookApiDTO](#) or the [API Reference](#).

The template payload is the string `My Webhook Template -- DATA: Action Details:`, plus the action details that are included in the action's data object. The variable `$action.details` is a reference to a field in the `ActionApiDTO` object that represents the current action. Your template can reference any of the fields in this DTO, starting with `action` as the object name. For example, `$action.createTime` gives you the time the action was created. For a full listing of the `ActionApiDTO` object, see [API Reference](#) or the [API Swagger UI](#).

## Sample Webhook Application

A webhook workflow sends a message to an application via HTTP. You express the message as a template that can include values from the action data in its payload. This template can express text, JSON, or any other payload that your application can accept.

You can use webhooks to send messages to a number of existing applications, including IBM CP4Integration, IBM CP4WatsonAIOps, Slack, Amazon Web Services, and others.

To deploy a simple example, and to test your webhook templates, you can implement a node.js server that receives the webhook message and prints out the template data. If you install this server on a machine in your network, then you can give its URL in the webhook workflow, and test your response to specific actions.

Following is a listing for a node.js web server that you can use.

```
let port = 9090;
const http = require("http");
console.log(`Starting server on port ${port}`);

http.createServer((request, response) => {
 request.setEncoding('utf8');
 console.log('REQUEST METHOD: ', request.method);

 let datStr = '';
 request.on('data', chunk => {datStr = datStr + chunk});
 request.on('end', () => {console.log('End of DATA: ', datStr)})
}).listen(port);
```

When you run this program, it prints a message to the console to say that it is running, and to identify the port it listens on.

When the server receives a message, it prints out the request method, and then prints out the message payload, as specified in the workflow's template field.

If you have configured an Automation Policy to use this workflow, then this server will log a message for each action that Turbonomic executes on an entity within the policy's scope.

## WebhookApiDTO

The WebhookApiDTO inherits from WorkflowAspect

### Required Parameters:

#### method

- **type:** string
- **description:** The http method used to make the request.
- **enum:** ['GET', 'POST', 'PUT', 'DELETE', 'PATCH']

#### url

- **type:** string
- **description:** The URL that HTTP request is made to.

### Optional Parameters:

#### template

- **type:** string
- **description:** The template for the body of request.

#### authenticationMethod

- **type:** string
- **description:** The authentication method to use for the request.
- **enum:** ['NONE', 'BASIC', 'OAUTH']

#### username

- **type:** string
- **description:** The username for the authenticated request.

#### password

- **type:** string
- **description:** The password for the authenticated request.

#### trustSelfSignedCertificates

- **type:** boolean

- **description:** If true, self-signed certificates will be trusted when using HTTPS connections. Defaults to 'false'.

#### headers

- **type:** array
- **description:** The request headers.

#### oauthData

- **type:** object
- **description:** Model to define the oAuth data.

##### Required Parameters:

- **clientId:** *string* The client id used for oAuth authorization.
- **clientSecret:** *string* The client secret used for oAuth authorization.
- **authorizationServerUrl:** *string* The URL of the authorization server.
- **grantType:** *enum* ["CLIENT\_CREDENTIALS"]

##### Optional Parameters:

- **scope:** *string* The oAuth scope.

## Working With Schedules

---

Schedules specify a period of time during which certain events can occur. Turbonomic uses the following types of schedules:

- Calendar Schedule

A calendar schedule is a setting that sets up a window of time when an [automation policy](#) takes effect. This policy can execute [non-parking actions](#) on public cloud or on-prem entities, or change settings that affect analysis and action generation.

- Parking Schedule

A parking schedule is a setting that sets up a window of time when parking actions (stop or start) are enforced on VMs and Database Servers (AWS RDS) in your public cloud environment. Click **Park** on the main menu to see a list of parkable VMs and Database Servers.

## Managing Calendar Schedules

A calendar schedule is a setting that sets up a window of time when an [automation policy](#) takes effect. This policy can execute [non-parking actions](#) on public cloud or on-prem entities, or change settings that affect analysis and action generation.

### Creating Calendar Schedules

1. Navigate to **Settings > Schedule**.
2. Click **New Schedule > Calendar Schedule**.
3. Type a name for the schedule.

Enter Schedule Name ✓ ✕ ✕

**RECURRENCE**  
Does not recur

**START DATE**  
Wed Jun 19 2019

**START TIME**      **DURATION**  
2:00 PM      1 hour(s)      0 minute(s)

**TIMEZONE**  
America/New\_York

SAVE

4. Set the recurrence for the schedule.

Choose whether the scheduled period occurs just once, or whether it repeats over time. The settings vary according to the recurrence you choose:

- Does Not Recur

This is a one-time schedule window. A non-recurring window has a start date, and no end date. The window starts on the day and time you specify, and remains open for the given duration.

- Daily

**RECURRENCE**  
Daily

**REPEAT EVERY**  
1 days

**START DATE**      **END DATE**  
Wed Jun 19 2019      None

Repeat this schedule every given number of days. For example, repeating 30 days is similar to repeating monthly, except it repeats by the count of days, not by the calendar month.

The schedule begins on the **Start Date**, and continues repeating until the **End Date**. If **End Date** is "None", the schedule repeats perpetually.

- Weekly



**RECURRENCE**

Weekly

**REPEAT EVERY** 1 weeks

**ON** Mo Tu Wd Th Fr Sa Su

**START DATE** Wed Jun 19 2019

**END DATE** None

Repeat this schedule every given number of weeks, on the week days you specify. For example, to repeat every weekend, set it to repeat every one week on Saturday and Sunday.

The schedule begins on the **Start Date**, and continues repeating until the **End Date**. If **End Date** is "None", the schedule repeats perpetually.

- Monthly

**RECURRENCE**

Monthly

**REPEAT EVERY** 1 months

**ON** First Saturday

**START DATE** Wed Jun 19 2019

**END DATE** None

Repeat this schedule every given number of months, to begin on a given day in the month. For example, you can schedule a maintenance window to begin on the first Saturday of each month.

The schedule begins on the **Start Date**, and continues repeating until the **End Date**. If **End Date** is "None", the schedule repeats perpetually.

5. Set the start time and duration.

These settings specify how long the scheduled window remains open. You set the duration in terms of hours and minutes. Using a duration instead of an end time removes ambiguities such as starting before midnight and ending after. However, you should make sure the duration is not longer than the recurrence.

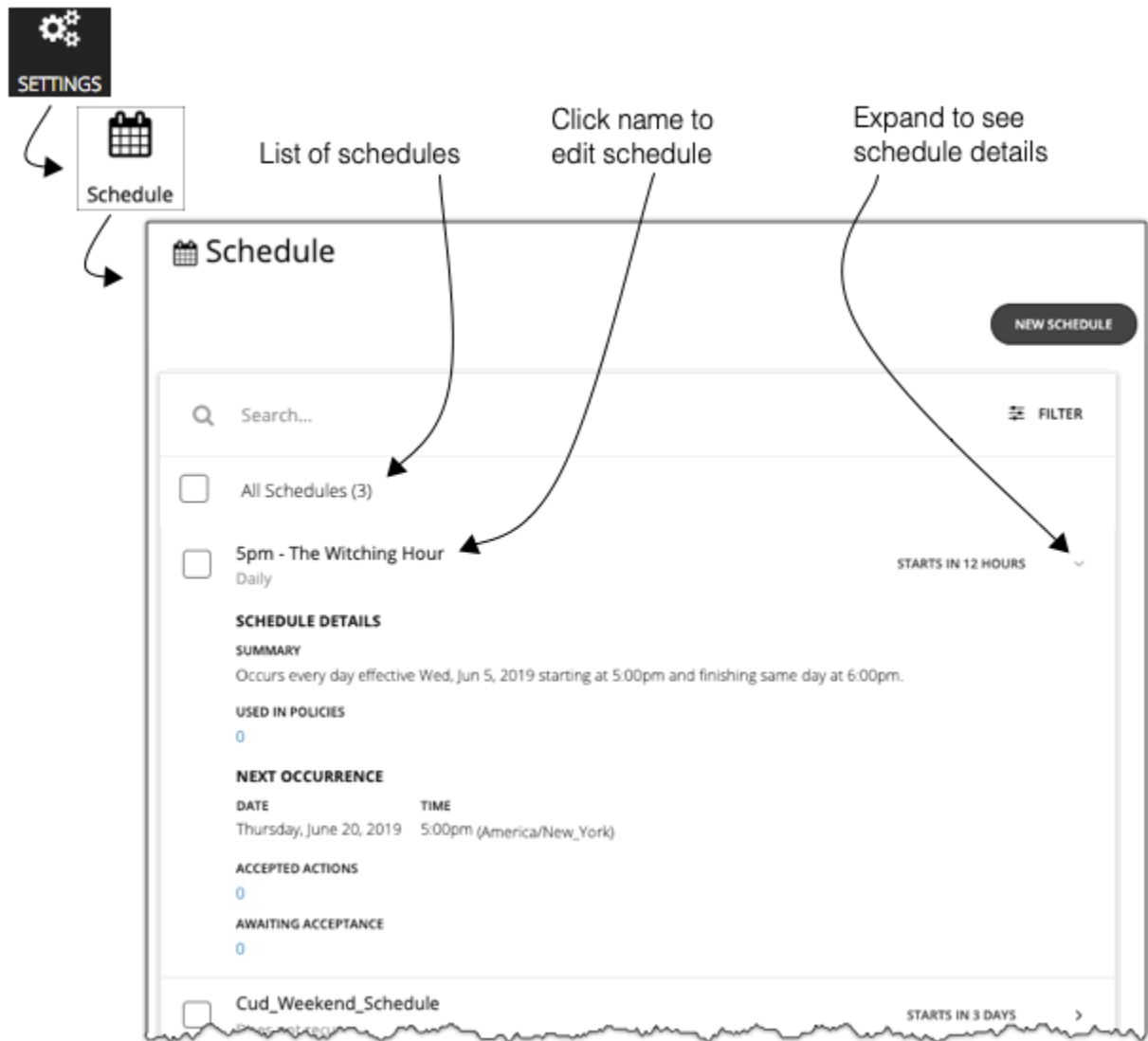
6. Set the time zone.

This gives a reference for the schedule's start time. Turbonomic uses that reference when it opens and closes the schedule window.

You see the same time zone setting no matter where you are located. Convert the schedule time to your local time to track when the schedule opens in your working day.

7. Save the schedule.

## Viewing Calendar Schedules



The Schedules page lists all the currently defined schedules. From this page you can:

- Select an entry to delete the schedule.
- Select an entry to defer the next occurrence.

Turbonomic calculates when the next scheduled window will open. If you want cancel the scheduled occurrence one time, you can select the schedule and defer the upcoming occurrence. This defers the schedule wherever it is applied. If the schedule is applied to more than one policy, this will defer all the policies that use this schedule. Before you defer a schedule, you should expand the details and review all the policies that use this schedule.

- Expand an entry to see schedule details

The details include a summary of the schedule definition, as well as:

- **Used in Policies**

The number of policies that use this schedule. Click the number to review the policies.

- **Next Occurrence**

When the schedule will next come into effect.

- **Accepted Actions**

How many scheduled actions have been accepted to be executed in the next schedule occurrence. Click the number for a list of these actions.

– **Awaiting Acceptance**

The number of Manual actions affected by this schedule that are in the Pending Actions list, and have not been accepted. Click the number for a list of these actions.

- Create new schedules.

## Deleting Calendar Schedules

Before you delete a schedule, you should view its details to make sure no policies use it. If you delete a schedule that is in use by any policies, Turbonomic disables the affected policies until you edit them to either:

- Apply a different schedule to the policy and save the change, or...
- Save the policy with no schedule

Saving with no schedule confirms that you intend for this policy to apply at all times. Because scheduled policies are for special cases, this is usually not what you intend. For example, a scheduled maintenance window can have aggressive action acceptance modes that you do not want to enable during peak hours. If you save the policy with no schedule, then the aggressive settings will take effect at all times.

Turbonomic posts a confirmation dialog before deleting a schedule that is currently in use.

## Managing Parking Schedules

A parking schedule is a setting that sets up a window of time when parking actions (stop or start) are enforced on VMs and Database Servers (AWS RDS) in your public cloud environment. Click **Park** on the main menu to see a list of parkable VMs and Database Servers.

### Creating Parking Schedules

1. Navigate to **Settings > Schedule**.
2. Click **New Schedule > Parking Schedule**.

The Create Schedule page displays.

### Create Schedule ✕

**SCHEDULE NAME \***

**DESCRIPTION**

**TIME ZONE**

America/New\_York ▼

**SCHEDULE RESOLUTION**

1 Hour ▼

HOURS UP: 0h    HOURS DOWN: 0h    STATE:  RUNNING    STOPPED    DO NOTHING

|     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|     | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| MON |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| TUE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| WED |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| THU |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| FRI |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SAT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| SUN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

3. Provide a name and description.

When attaching a parking schedule to one or more entities, you will see the *name* of the schedule. It is therefore important to use a name that will help you recognize each schedule and choose the appropriate one.

4. Set the time zone.

This gives a reference for the schedule's start time. Turbonomic uses that reference when it opens and closes the schedule window.

You see the same time zone setting no matter where you are located. Convert the schedule time to your local time to track when the schedule opens in your working day.

5. Set the schedule resolution.

By default, each block in the schedule configuration section corresponds to an hour of the day. You can change this to 15-minute increments to give you more flexibility with scheduling. For example, with 15-minute increments, you can enforce parking actions between 8:15 and 20:45. This is not possible with the default schedule resolution.

6. Set a parking schedule.

Parking schedules recur every week, on the days and times that you specify. These schedules include the following states that define the parking policies that you can enforce.

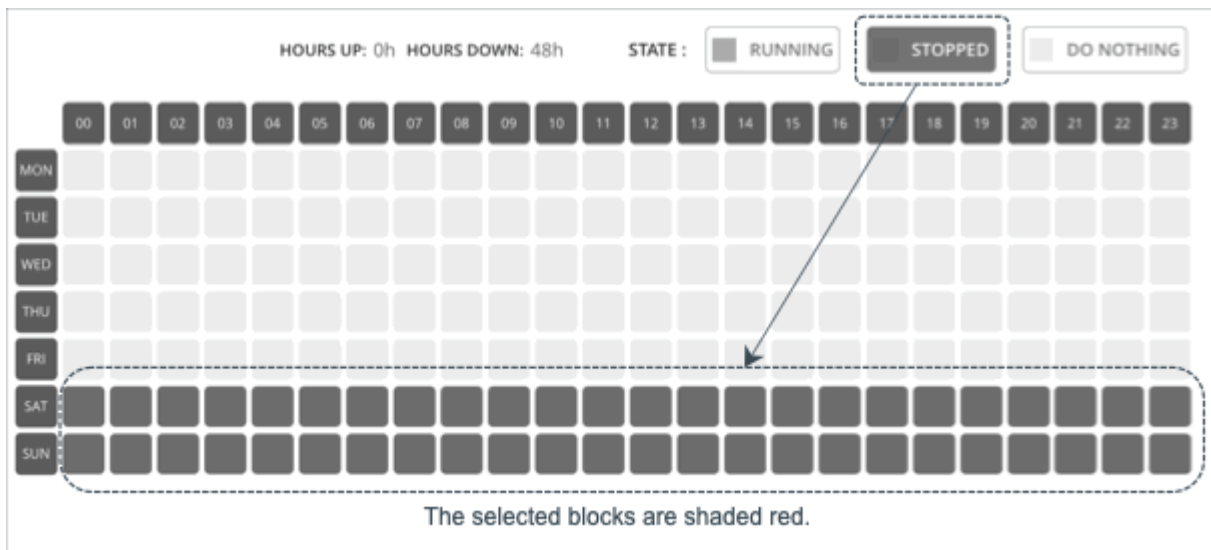
| State           | Policy                                                                                                                |
|-----------------|-----------------------------------------------------------------------------------------------------------------------|
| Running (green) | A parkable entity must be running on the selected days and times.<br>Turbonomic will start the entity immediately if: |

| State             | Policy                                                                                                                                                                                                                                                                                                  |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>The entity is not running at the beginning of the schedule.</li> <li>A user stops the entity via the cloud provider.</li> </ul>                                                                                                                                  |
| Stopped (red)     | <p>A parkable entity must not be running on the selected days and times.</p> <p>Turbonomic will stop the entity immediately if:</p> <ul style="list-style-type: none"> <li>The entity is running at the beginning of the schedule.</li> <li>A user starts the entity via the cloud provider.</li> </ul> |
| Do Nothing (gray) | Do not enforce parking actions on the selected days and times. This is the default state for all the blocks.                                                                                                                                                                                            |

Basic steps:

- Click either the **Running** (green) or **Stopped** (red) button.
- Click the blocks for the days and times the action will be enforced. To select multiple contiguous blocks, click and hold, and then drag in any direction.
- To undo your selections, click the **Do Nothing** (gray) button and then click the affected blocks.

In the following example, stop actions are enforced every Saturday and Sunday. No parking actions are enforced on all the other days of the week.



7. Click **Create**.

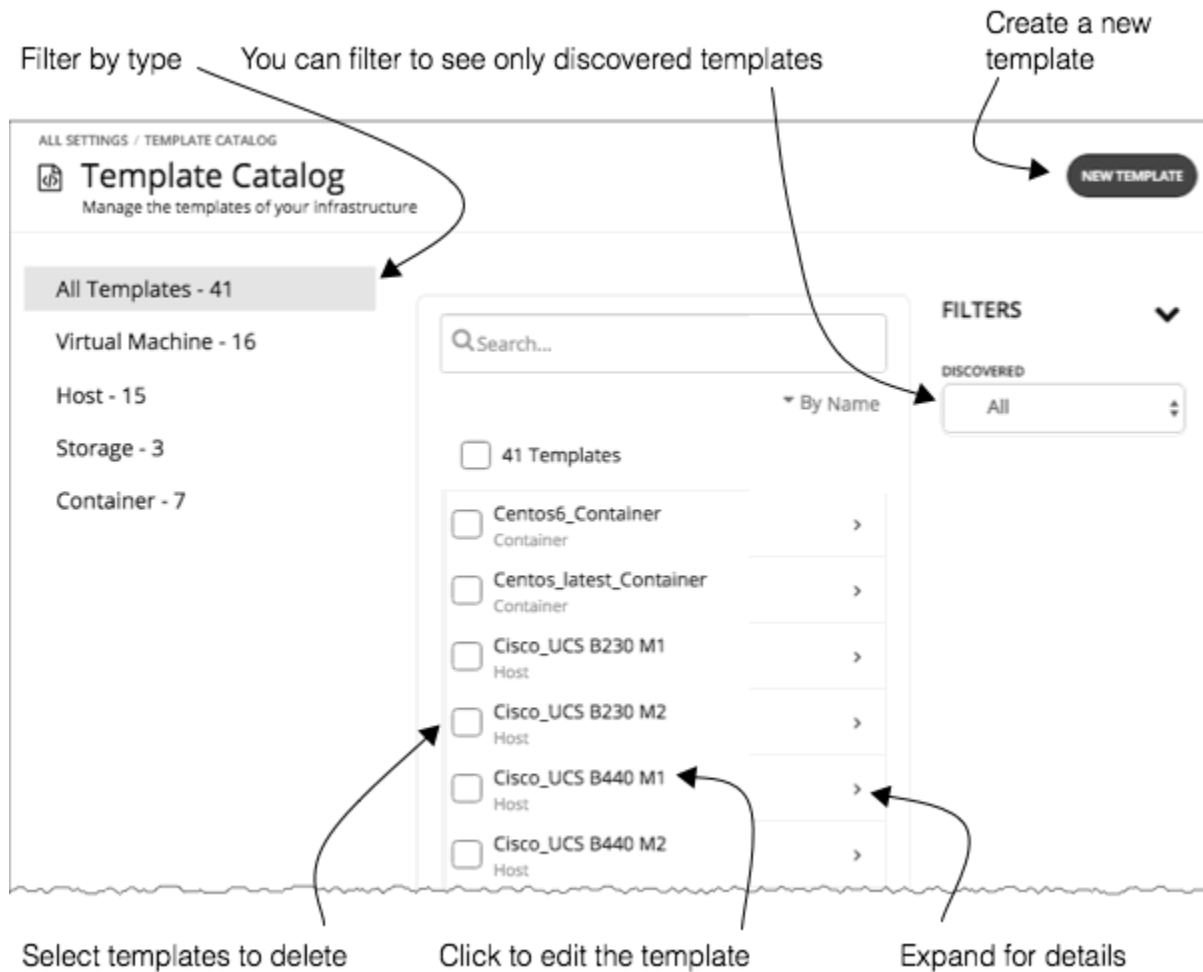
The schedule can now be attached to the parkable entities listed in the Parking page. For details about the Parking page, see [Park: Stop or Start Cloud Resources](#).

## Deleting Parking Schedules

Only unattached parking schedules can be deleted.

- Select one or several unattached schedules to delete.
- Click the delete button at the top-right section of the page.

## Templates: Resource Allocations for New Entities



Turbonomic uses templates to describe new entities that it will deploy in your environment or in plans. The templates specify resource allocations for these entities. For example, you can run a plan that adds new VMs to a cluster. If you add ten copies of a template, then the plan places ten new VMs that match the resource allocation you have specified for the given template. For your cloud environment, you can see templates to match the instance types in your cloud accounts and subscriptions.

A VM template definition can include one or more images that Turbonomic uses to deploy the VM in your environment. The image identifies the actual deployment package, including a path to the physical files (for example an OVA).

As you deploy an instance of a VM template, Turbonomic chooses the best image for that instance.

The Template Catalog shows all of the templates that have been specified or discovered for your installation of Turbonomic. From this page, you can also create new templates and edit existing ones.

### Creating Templates

Templates specify the resources for entities that Turbonomic can deploy in your environment, or in plans.

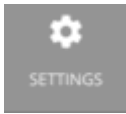
A VM template definition can include one or more images that Turbonomic uses to deploy the VM in your environment. The image identifies the actual deployment package, including a path to the physical files (for example an OVA).

The Template Catalog shows all of the templates that have been specified or discovered for your installation of Turbonomic. From this page, you can also create new templates and edit existing ones.

## Creating and Editing Templates

To create a new template, navigate to the Template Catalog and click **NEW TEMPLATE**. To edit a template, click the template's name. When you create a new template, the first step is to choose the entity type.

1. Navigate to the Settings Page.



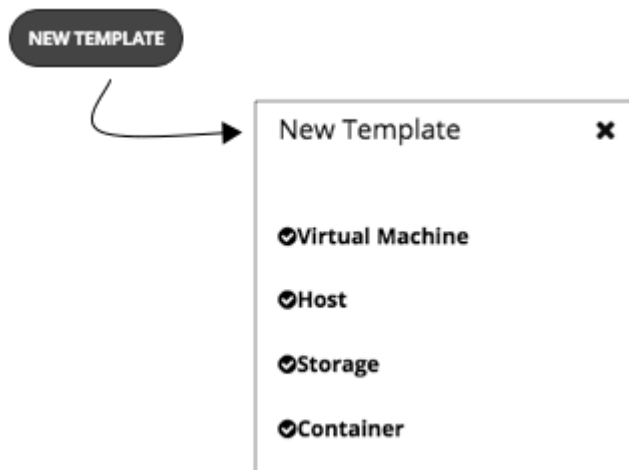
2. Choose Templates.



3. Create or edit a template

To create a new template, navigate to the Template Catalog and click **NEW TEMPLATE**. To edit a template, click the template's name.

4. If you're creating a new template, choose the entity type.



5. Make the settings for your template.

For each type of template, you set allocations for different resources. You can make templates of the following types:

- Virtual Machine
- Host
- Storage
- Container

6. Make the settings for your template, and then save your changes.

When the template window opens, it displays the most common resource settings. You can expand the settings to see the full collection for that template type.

7. Save your changes.

After you have made your settings and named the template, click **CREATE** or **SAVE**.

## VM Template Settings

A VM template describes the resource allocation that you want to provide for a type of VMs. When Turbonomic deploys the associated VM to your environment or in a plan, it uses these values to determine the size of the VM. Turbonomic uses the Size settings to calculate the best placement for a VM of this type.

A VM template can optionally include an image description. When Turbonomic uses the template to deploy a VM to your environment, it uses the image to access the actual bits that install as the VM instance.

### Note:

Turbonomic generates a special template called *headroomVM*, which it uses to calculate cluster headroom. The Template Catalog shows the template as editable, but you should not edit it because Turbonomic will overwrite your changes the next time it generates the template.

## VM Size

- CPU

The virtual CPUs assigned to the VM. Specify the number of **Cores** and the **VCPU** clock speed – Turbonomic multiplies these values to calculate the host CPU resources it will allocate when placing the VM.

The **Utilization** value sets the percentage of allocated CPU that the placed VM will consume. To ensure the host has left over resources for infrastructure tasks, you should assign less than 100%.

- Memory

The amount of memory to allocate for the VM, in MB.

The **Utilization** value sets the percentage of allocated memory that the placed VM will consume. To ensure the host has left over resources for infrastructure tasks, you should assign less than 100%.

Note that you should never allocate less memory than is required for the VM's guest OS.

- Storage

The storage resources to allocate for this VM.

- **disk/rdm** – If you choose **rdm**, then the VM can use VMware Raw Device Mapping for its storage.
- **IOPS** – The capacity for IO operations you give the VM for this datastore.
- **Size** – The amount of storage capacity, in GB.

The **Utilization** value sets the percentage of allocated memory that the placed VM will consume. To ensure the storage has left over resources for infrastructure tasks, you should assign less than 100%.

Note that you can allocate multiple datastores to the VM.

- Network

The amount of the host's network throughput to assign to the VM, in Mb/s.

- IO

The amount of throughput on the host's IO bus to assign to the VM, in Mb/s

## Host Template Settings

Host templates describe models of physical hosts that you can deploy in the on-prem datacenter. As part of capacity planning, you might want to see how to replace your current hosts with different models. To do that, you create templates to represent the hosts you want, and then use those templates when running hardware replacement plans.

The host template is a collection of these settings:

- CPU



The processor for this host model. Note that CPU size and speed are not the only factors to determine processing power. To address this, you can specify the host CPU in the following ways:

- Select from Catalog



When you enable **Select from Catalog**, you can open up a catalog of CPU models that Turbonomic uses to map the model to an effective capacity for the CPU.

- Cores and CPU Speed



When you disable **Select from Catalog**, you can specify the number of **Cores** and the **CPU** clock speed – Turbonomic multiplies these values to calculate the host CPU resources.

- Memory

The amount of memory to allocate for the VM, in MB.

- Network

The host's network throughput, in MB/s.

- IO

The host's IO bus throughput, in MB/s

- Price

If you know the price of the host model that you're specifying for the template, you can enter it here. When running a plan, Turbonomic can use the price to calculate costs or savings when adding or removing host machines in an on-prem datacenter.

## Selecting CPUs from the Catalog

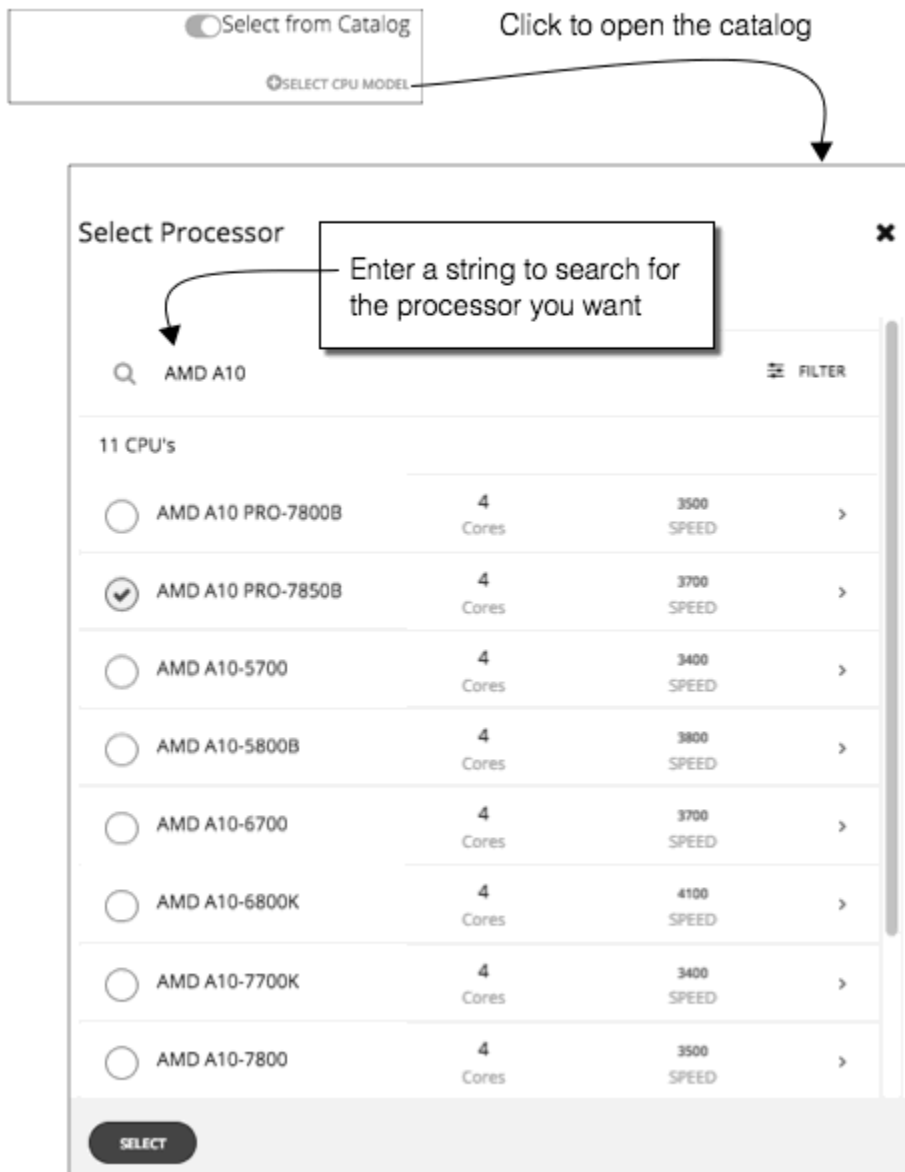
CPU processor speed is not necessarily an effective indicator of CPU capacity. For example, processor architecture can make a slower CPU have a greater effective capacity. Newer models of machines can often have fewer cores or less clock speed, but still have a higher effective capacity. This can affect planning in two ways:

- When planning hardware replacement, the plan knows the template's effective capacity. This means the plan knows how to best place workloads on the new hardware.
- For already deployed hosts, Turbonomic discovers the effective capacity and uses that information when calculating workload placement.

To build the catalog of CPU capacity, Turbonomic uses benchmark data from [spec.org](http://spec.org). When you set up the CPU for a host template, you can search this catalog for the processor you want, and set it to the template.

### Note:

Turbonomic also uses the effective processor capacity when calculating workload placement in real-time. For more information, see [Effective CPU Capacity](#).



## HCI Host Template Settings

HCI host templates describe models of physical hosts that support participation in a vSAN. Along with the host compute specifications, you also include specifications for storage capacity and redundancy (RAID level and failover). You can use these templates to plan for changes to your vSAN capacity.

### Note:

For Hyper-V environments, if you run a Hardware Replace plan that replaces hosts with HCI Host templates, the results can be inconsistent or the plan can fail to place all the VMs in the plan scope. This typically occurs when Turbonomic detects a configuration issue with VMM or Hyper-V. As a result, Turbonomic treats the VMs as not controllable and will not attempt to place them.

The HCI Host template is a collection of these settings:

- CPU

The processor for this host model. Note that CPU size and speed are not the only factors to determine processing power. To address this, you can specify the host CPU in the following ways:

- Select from Catalog



When you enable **Select from Catalog**, you can open up a catalog of CPU models that Turbonomic uses to map the model to an effective capacity for the CPU.

- Cores and CPU Speed



When you disable **Select from Catalog**, you can specify the number of **Cores** and the **CPU** clock speed – Turbonomic multiplies these values to calculate the host CPU resources.

- Memory

The amount of memory to allocate for the VM, in MB.

- Network

The host's network throughput, in MB/s.

- IO

The host's IO bus throughput, in MB/s

- Storage

The capacity for this storage.

– **IOPS** – The effective IOPS capacity.

– **Size** – Raw storage capacity, in GB. A plan that uses this template will compute the effective storage capacity.

- Redundancy

The redundancy method for this storage on the virtualized SAN. This combines the RAID level and the number of host failures to tolerate.

- Price

If you know the price of the host model that you're specifying for the template, you can enter it here. When running a plan, Turbonomic can use the price to calculate costs or savings when adding or removing host machines in an on-prem datacenter.

## Selecting CPUs from the Catalog

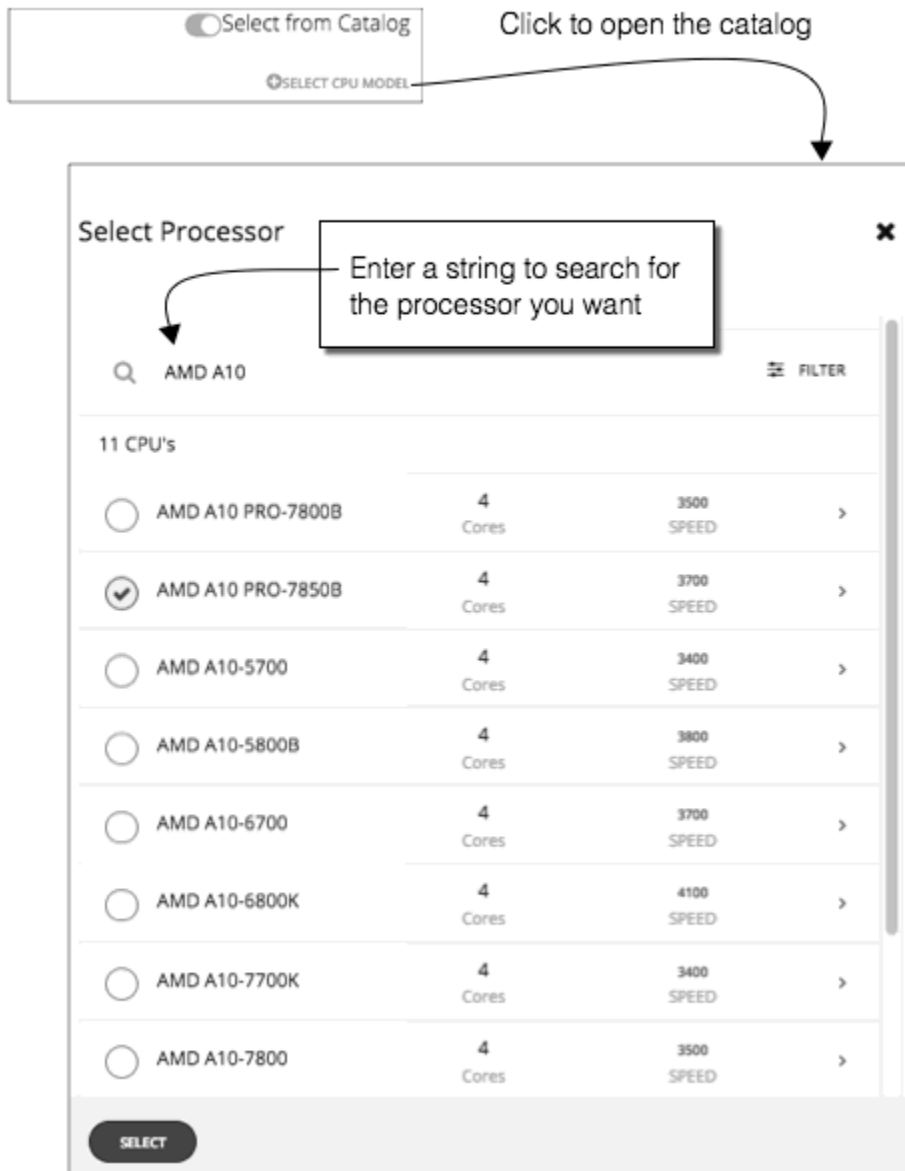
CPU processor speed is not necessarily an effective indicator of CPU capacity. For example, processor architecture can make a slower CPU have a greater effective capacity. Newer models of machines can often have fewer cores or less clock speed, but still have a higher effective capacity. This can affect planning in two ways:

- When planning hardware replacement, the plan knows the template's effective capacity. This means the plan knows how to best place workloads on the new hardware.
- For already deployed hosts, Turbonomic discovers the effective capacity and uses that information when calculating workload placement.

To build the catalog of CPU capacity, Turbonomic uses benchmark data from spec.org. When you set up the CPU for a host template, you can search this catalog for the processor you want, and set it to the template.

### Note:

Turbonomic also uses the effective processor capacity when calculating workload placement in real-time. For more information, see [Effective CPU Capacity](#).



## Storage Template Settings

Storage templates describe models of storage that you can deploy in the on-prem datacenter. As part of capacity planning, you might want to see how to replace your current storage with different models. To do that, you create templates to represent the storage you want, and then use those templates when running hardware replacement plans.

The storage template is a collection of these settings:

- Storage
  - The capacity for this storage.
    - **IOPS** – The capacity for IO operations on this storage.
    - **Size** – The amount of storage capacity, in GB.
- Price

If you know the price of the storage model that you're specifying for the template, you can enter it here. When running a plan, Turbonomic can use the price to calculate costs or savings when adding or removing storage in an on-prem datacenter.

## Billing and Costs

As you work with Turbonomic, you can set up costs that Turbonomic uses in its calculations. This setup includes:

- Reserved Instance Settings

To recommend placing workloads on instance types that take advantage of discounted pricing, Turbonomic uses the real pricing plans that are available to the targets public cloud accounts. Setting up a purchase profile adds even more detail to the pricing structure that Turbonomic uses in its calculations.

- Price Adjustment

Cloud service providers can offer their own price lists, including special costs for services or discounts for workloads. However, Turbonomic does not discover these adjustments. For example, to reflect any discounted prices in the Turbonomic display and in Turbonomic analysis, you must manually configure those discounts. In Turbonomic, you configure such discounts via **Price Adjustments** for specific billing groups in your cloud environment.

- Currency

By default, Turbonomic uses the dollar symbol (\$) when displaying the costs and savings that it discovers or calculates for your cloud workloads. You can set a different symbol to match your preferred currency. For example, if your cloud provider bills you in euros, change the currency symbol to €.

## Reserved Instance Settings

The screenshot shows the 'Reserved Instance Settings' configuration page. It is organized into two main sections: 'AWS PROFILE' and 'AZURE PROFILE'.  
Under 'AWS PROFILE':  
- 'OFFERING CLASS' has two radio buttons: 'Standard' (selected) and 'Convertible'.  
- 'TERM' has two radio buttons: '1 Year' (selected) and '3 Year'.  
- 'PAYMENT' has three radio buttons: 'All Upfront' (selected), 'Partial Upfront', and 'No Upfront'.  
Under 'AZURE PROFILE':  
- 'TERM' has two radio buttons: '1 Year' (selected) and '3 Year'.  
At the bottom right, there are two buttons: 'RESET DEFAULTS' and 'APPLY SETTINGS'.

To recommend placing workloads on instance types that take advantage of discounted pricing, Turbonomic uses the real pricing plans that are available to the targets public cloud accounts. Setting up a purchase profile adds even more detail to the pricing structure that Turbonomic uses in its calculations.

A purchase profile determines the costs that Turbonomic will use for all discount purchase decisions in your environment. As it sees opportunities to move workloads to another term, Turbonomic determines

the costs based on the profile, and includes cost information in action descriptions. Turbonomic also uses this information to calculate projected changes in costs.

Note that the settings you configure apply to your global public cloud environment.

To set up a profile, navigate to **Settings > Billing and Costs**, and display the **RESERVED INSTANCE SETTINGS** tab. Then make the settings for your purchase profile:

- Offering Class

For AWS environments, choose the offering class that corresponds to the RI types that you typically use in your environment.

- Term

For AWS and Azure environments, choose the payment terms you contract for your discounts. TERM can be one of **1 Year** or **3 Year**. Typically, longer term payment plans cost less per year.

- Payment

The payment option that you prefer for your AWS RIs:

- All Upfront – You make full payment at the start of the RI term.
- Partial Upfront – You make a portion of the payment at the start of the term, with the remain cost paid at an hourly rate.
- No Upfront – You pay for the RIs at an hourly rate, for the duration of the term.

When you are satisfied with your RI Purchase Profile settings, click **APPLY SETTINGS**. Or to reset the form, click **RESET DEFAULTS**.

## Price Adjustments

Cloud service providers can offer their own price lists, including special costs for services or discounts for workloads. However, Turbonomic does not discover these adjustments. For example, to reflect any discounted prices in the Turbonomic display and in Turbonomic analysis, you must manually configure those discounts. In Turbonomic, you configure such discounts via **Price Adjustments** for specific billing groups in your cloud environment.

Turbonomic applies these price adjustments to:

- Costs for workload template families, including:
  - Compute
  - Discount Compute
- Costs for services, including:
  - Bandwidth
  - VM Licenses
  - AWS CloudWatch
  - AWS DynamoDB
  - And others

Note that in AWS environments, Turbonomic does not apply any discounts or other price adjustments to Spot Compute costs.

The general steps to configure a price adjustment are:

- Create the price adjustment:

- Specify the adjustment scope

To do this, you choose which cloud service provider is giving you the adjustment, and then choose a billing group to set the scope of the adjustment.

- Choose the Type

The price adjustment can be a Discount or an Increase. In most cases you will specify discounts for the price adjustment. While this sets the type for the overall adjustment, you can override the type for specific line items.

- Specify a Price Adjustment setting

The Price Adjustment is the overall adjustment that your cloud service provider offers for the billing groups in your current scope. For example, AWS might offer you a 10% discount for a given account. For that billing group, you would specify a 10% Discount for the Price Adjustment setting.

- Specify Price Overrides (AWS only)

While your service provider might offer a general price adjustment for the billing group you chose, it might also offer further discounts for select services or template families. Or it might offer discounts for some template families, but price increases for some other services. You can configure these differences as Price Overrides.

**Note:**

Turbonomic uses the adjustments that you configure to display costs in the user interface. However, the values for hourly cost per entity, total hourly cost, total monthly cost, or total yearly cost can show inaccuracies on the order of a fraction of a percent. This is due to rounding when calculating the adjusted cost per entity.

## Creating a Price Adjustment

A price adjustment configures adjusted workload pricing that you have negotiated with your Cloud Provider. After you configure an adjustment, Turbonomic applies it to pricing in the affected cloud scope.

To create a price adjustment in Turbonomic, you identify the adjustment's scope – the subscriptions or billing families the adjustment applies to – and then set the type and percentage for the price adjustment. This specifies an overall adjustment for the workloads that fall within the billing group. For AWS, you can later drill into the adjustment to specify overrides for specific template families or services.

**Notes:**

- To use a price adjustment with a given billing group, you must increase the memory allocated to the VM that hosts your Turbonomic instance. Turbonomic requires that you provide a minimum amount of memory when you install the product. To use price adjustments, IBM recommends that you increase the allocated memory as follows:
  - For the first price adjustment assigned to one or more billing groups, increase by 4 GB.
  - For each subsequent price adjustment assigned to one or more billing groups, increase by an additional 1 GB.
- Whenever you add, edit, or remove a Price Adjustment that is in use, you must allow sufficient time for Turbonomic to fully discover all of the affected environment, and to propagate the changes throughout that environment. In an average environment, this can take up to 30 minutes. As an alternative, you can manually execute rediscovery for the affected cloud subscription or account.

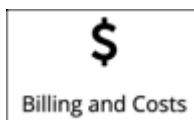
To create a price adjustment:

1. Navigate to the Settings Page.



Click to navigate to the Settings Page. From there, you can perform a variety of configuration tasks.

2. Choose Billing and Costs.



Click to navigate to the Billing and Costs page.

3. Display the PRICE ADJUSTMENT tab.

Click the **PRICE ADJUSTMENT** tab to see all of the adjustments that have been configured for your environment. In this list you can:

- Click an entry to see details and edit the adjustment
- Select an entry to delete the adjustment
- Create new price adjustments

4. Create the price adjustment.

First click **NEW PRICE ADJUSTMENT**, then specify the following settings to configure a price adjustment:

- Give the adjustment a name.
- To set the scope for this adjustment, choose its Billing Groups.

Click in the **BILLING GROUPS** field to display the Billing Groups fly-out.

In the Billing Groups fly-out, choose the cloud service provider you want to work with and then choose the billing group for the scope of this adjustment.

A Billing Group is a set of cloud service provider accounts that are consolidated into a single billing schedule. Billing group details depend on your service provider:

- **AWS:** To consolidate billing, AWS supports billing families of AWS accounts, where there is a *master* account and other *member* accounts. Turbonomic lists each billing family as a billing group. You can choose a billing family to set the scope of this adjustment.

After you have chosen your billing group, click **SAVE** to return to the Add New Price Adjustment fly-out.

- Set the Type for this price adjustment – Choose either **Discount** or **Increase**.
- Specify a percentage of adjustment as the Price Adjustment.

Enter the percentage in the **PRICE ADJUSTMENT** field. The acceptable value depends on the type of adjustment:

- For a discount: 0 - 99.99%
- For an increase: 0 - 999.99%

This is the general percentage of adjustment (increase or discount) for the current scope. For any costs within the adjustment scope, Turbonomic will apply this percentage as it calculates the optimal workload capacity and placement.

**Note:**

If you set an overall adjustment of 0%, then Turbonomic enforces a Type setting of Discount. The end result is the same, because an increase or a discount of 0% is the same.

5. (AWS only) Specify any price overrides for this price adjustment.

The PRICE ADJUSTMENT percentage you just specified applies as a default in the adjustment scope. However, you might have negotiated different prices for specific services or template families in your cloud environment. To configure these special prices, click **PRICE OVERRIDES** to open the Cloud Cost Adjustment fly-out.

For details, see [AWS Price Override](#).

6. Save your work.



After you have configured the price adjustment, click **SAVE**.

## AWS Price Override

| Cloud Cost Adjustment [AWS] - My AWS Discount |            |                    |            |                       |                        |                       | ✕          |
|-----------------------------------------------|------------|--------------------|------------|-----------------------|------------------------|-----------------------|------------|
|                                               |            |                    |            |                       |                        |                       | DOWNLOAD ▾ |
| SERVICES                                      | TYPE       | PRICE ADJUSTMENT % | OVERRIDE % | ORIGINAL RATE (LINUX) | EFFECTIVE ADJUSTMENT % | ADJUSTED RATE (LINUX) |            |
| AWS CloudTrail                                | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |
| AWS CloudWatch                                | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |
| AWS Developer Sup...                          | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |
| AWS DynamoDB                                  | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |
| ^ AWS EC2 Compute *                           | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |
| ^ c5d *                                       |            |                    |            | —                     |                        | —                     |            |
| ▾ c5d.9xlarge...*                             | Discount ▾ | 10 %               | 15 %       | —                     | 15 %                   | —                     |            |
| ▾ c5d.18xlarge...                             | Discount ▾ | 10 %               | %          | —                     | 10 %                   | —                     |            |

To override the PRICE ADJUSTMENT setting for AWS Billing groups, Turbonomic analysis can use settings for different services that AWS provides to your accounts.

In AWS, you can set up a billing family that includes a *master* account and a given set of *member* accounts. Turbonomic treats the AWS Billing family as a Billing Group. For more information about billing families and accounts, see [AWS Billing Families](#).

Assume you have configured a price adjustment with a discount of 10% for a billing family, to match the overall discount that AWS offers you for that scope. But then assume the account includes extra discounts for some of the services your billing families provide. Then you can create overrides to add the extra discounts to those services.

Turbonomic uses the adjusted costs in its analysis as it calculates actions. For example, assume a price adjustment of 10% for a billing group, and a discount of 20% for the M4.Large family of templates. As Turbonomic places a workload, it will consider both the template capacity and the template cost. Even if an M4 template is larger than the workload actually needs, the M4 template could be less expensive because of the added discount. In that case, Turbonomic will place the workload on the less expensive template.

### Note:

The Cloud Cost Adjustment table lists the services that are available to you for the AWS Billing family that you have set up as the discount scope. The services this table displays depend on whether the billing family uses the given service, and whether there is any recorded cost at the time that you display the table. For this reason, under some circumstances you might see different services listed in the table.

Under all circumstances, the table lists the services, AWS EC2 Compute, AWS EC2 Reserved Instance, and AWS RDS.

Also, for the Cloud Cost Adjustment table to display CSP Cost and Effective Cost, you must have created a Cost and Usage report in AWS, and you must store it in an S3 bucket.

In the Cloud Cost Adjustment table, you can perform the following:

- Override the price adjustment for a service or template family.

To add an override, choose the line item for a service, or expand the row for a template family and:

- Set the Type. Double-click and then choose **Discount** or **Increase**. Press **Enter** to confirm your setting.
- Specify the percentage for this override, and then press **Enter** to confirm your override. The value you enter here is an absolute value for the discount or increase Turbonomic will apply for this line item.

When you're done setting these overrides, click **Save**.

- To remove all overrides and revert back to the PRICE ADJUSTMENT Discount, click **CLEAR ALL OVERRIDES**.
- To download a report of the discounts for each service, click **DOWNLOAD** and choose CSV or PDF.

The table lists the following information about your discounts:

- SERVICES

The different cloud services to which you can set an override discount. To see individual workload templates:

- For Azure, expand **Virtual Machines**
- For AWS, expand **AWS EC2 Compute** or **EC2 Reserved Instance**

- TYPE

Whether this price adjustment will be an increase or a discount. By default, this field shows the setting that you have made for the Price Adjustment. However, you can change it as an override for an individual entry.

- PRICE ADJUSTMENT %

The percentage that you have specified for the Price Adjustment setting. This is the general adjustment that Turbonomic applies by default to the given service.

- OVERRIDE %

If you have entered a value, this is the price adjustment Turbonomic applies to the given service.

- ORIGINAL RATE (LINUX)

The Cloud Service Provider's cost for VM templates, per hour. To see these costs, expand the workload services to show specific templates. The cost assumes no charge for the OS license, as though the VM runs Linux.

- EFFECTIVE ADJUSTMENT %

The actual adjustment for the given service.

- ADJUSTED RATE (LINUX)

The discounted cost for VM templates, per hour. To see these costs, expand **Virtual Machines** to show specific templates. The cost assumes no charge for the OS license, as though the VM runs Linux.

## AWS Billing Families



As you configure AWS targets, Turbonomic discovers AWS accounts that are consolidated into *billing families*. A billing family has one *master* account, and zero or more *member* accounts. By recognizing billing families, Turbonomic more accurately calculates cloud investments and savings, and makes more accurate recommendations for RI coverage.

In the Targets user interface, master accounts appear in bold, with a star next to them. You can expand the account entry to see the related member accounts. If you expand the entry for a member account, then the related accounts includes the family master, indicated by a star.

For RI purchases, different accounts in a billing family can share the same RI resources. At the same time, accounts in other billing families cannot use those RIs. This adds flexibility to your RI coverage, while maintaining order over the billing.

In Turbonomic, if you enable Billing Family Recognition, then you can see the billing family master and member accounts in the Targets user interface, and Turbonomic can recommend proper RI purchases within the correct billing families.

To enable Billing Family Recognition, ensure the following as you configure your AWS targets:

- Use the proper role for each AWS target

To properly discover billing family information for a target, you must give Turbonomic credentials for an AWS role that includes the permission, `organizations:DescribeOrganization`. With that permission, Turbonomic can:

- Discover master accounts and member accounts in different billing families
- Display the account names in the user interface
- Discover billing information for each family and account
- Recommend RI actions that respect billing family boundaries

- Configure targets for the complete billing family

One billing family can consolidate a number of AWS accounts. For Turbonomic to include these accounts in its analysis, you must configure each one as a separate target. If you do not configure all the accounts in a billing family, then Turbonomic cannot discover complete billing information for that family, and its analysis will be based on incomplete information.

Turbonomic displays member accounts that have been configured as targets in regular text. For members that Turbonomic discovers but have not been configured as targets, Turbonomic displays their names in grayed text.

If you have enabled Billing Family Recognition, you should keep the following points in mind:

- Billing families can grow

Turbonomic regularly checks the membership of your billing families. If it discovers a new member account, it adds that account to the list of members. If you have already configured the account as a target, then Turbonomic includes the new member in its analysis of billing families. If the new member is not already a target, then Turbonomic lists the new member in grayed text.

- You can configure discounts per billing family

Turbonomic includes a feature to set a discount for a billing group, and to override that discount for specific template families within that scope. For more information, see [Cloud Discounts](#) and [Discount Override: AWS](#).

- You might see master accounts that have no member accounts

AWS treats every account you create as a part of a billing family. Assume you created an account, but you had no reason to consolidate its billing with any other accounts. In that case, the account appears in the Turbonomic user interface as a master account, but it has no member accounts.

## Currency Settings

By default, Turbonomic uses the dollar symbol (\$) when displaying the costs and savings that it discovers or calculates for your cloud workloads. You can set a different symbol to match your preferred currency. For example, if your cloud provider bills you in euros, change the currency symbol to €.

To change the currency symbol, go to **Settings > Billing and Costs** and then click the **Currency** tab.

Turbonomic saves your preference in the local storage of the browser that you used to access the user interface. It reverts to the default symbol if you use another browser or view the user interface in incognito/private mode.

Currency symbols are for display purposes only. Turbonomic does not convert monetary amounts when you switch symbols.

## Administrative Tasks

---

To perform Turbonomic administrative tasks, you will navigate to different pages from **Settings**. The different tasks you can perform for Turbonomic include:

- [Managing User Accounts](#)

Create and manage user accounts for Turbonomic.

- [Maintenance: Logging, and Troubleshooting](#)

Perform general tasks such as setting log levels or sending troubleshooting data to Technical Support.

- [Viewing the Update page](#)

See information about your current version. Note that this page is not available in SaaS instances of Turbonomic.

- [License Configuration](#)

Review the status of your current license, and apply any license upgrades.

## Managing User Accounts

As an administrator, you specify accounts that grant users specific access to Turbonomic. User accounts determine the following for a given user login:

- User Authentication

To configure an account, you set the type of authentication the account will use:

- Local User – Configure the username and password and save those credentials on the Turbonomic server.
- External User – Single user accounts that authenticate through Single Sign-on (SSO) or through Microsoft Active Directory (AD).
- External Group – A group of user accounts that authenticate through SSO or AD.

- User Authorization

Properties that determine the range of access and features for a given user:

- Role – Access to specific Turbonomic features
- Scope – How much of the environment this user can manage

As you configure user accounts, you can set up access to specific clusters in your environment. You can even set up accounts for tenant customers, and only show them the virtual workloads they own in their specific virtual datacenters.

### Important:

For self-hosted Turbonomic instances, you can configure Turbonomic to use SSO authentication. When SSO is enabled, Turbonomic only permits logins via the SSO IdP. Whenever you navigate to your Turbonomic installation, it redirects you to the SSO Identity Provider (IdP) for authentication before displaying the Turbonomic user interface.

Before you enable SSO for your Turbonomic installation, *you must configure at least one SSO user with Turbonomic administrator privileges*. If you do not, then once you enable SSO you will not be able to configure any SSO users in Turbonomic. To authorize an SSO user as an administrator, use **EXTERNAL AUTHENTICATION** to do one of the following:

- Configure a single SSO user with administrator authorization.

Add an external user. The username must match an account that is managed by the IdP.

- Configure an SSO user group with administrator authorization.

Add an external group. The group name must match a user group on the IdP, and that group must have at least one member.

For information about configuring SSO user groups in SAML, see [Configuring a Group for SSO Authentication](#). For information about configuring SSO authentication for Turbonomic, see [Single Sign-On Authentication](#).

To work with Turbonomic accounts:

1. Navigate to the Settings Page.



Click to navigate to the Settings Page. From there, you can perform a variety of Turbonomic configuration tasks.

2. Choose User Management.



## User Management

Click to navigate to the User Management Page.

Manage local or external authentication

The screenshot shows the 'User Management' interface. At the top, there are tabs for 'LOCAL USERS' and 'EXTERNAL AUTHENTICATION'. A 'NEW LOCAL USER' button is in the top right. Below the tabs is a list of 'All Local Users' with roles like Administrator, Advisor, Automator, Site Administrator, and Deployer. To the right is a search and filter section with a search bar and a 'FILTER' button. Below the search bar, there are checkboxes for '8 Users', 'administrator', 'Alice', and 'Bob'. Arrows point from the text 'Manage local or external authentication' to the tabs, 'List of accounts' to the user list, and 'Click a name to edit the account Select an account to delete it.' to the user names in the list.

This page lists all the user accounts that you currently have configured for Turbonomic. You can:

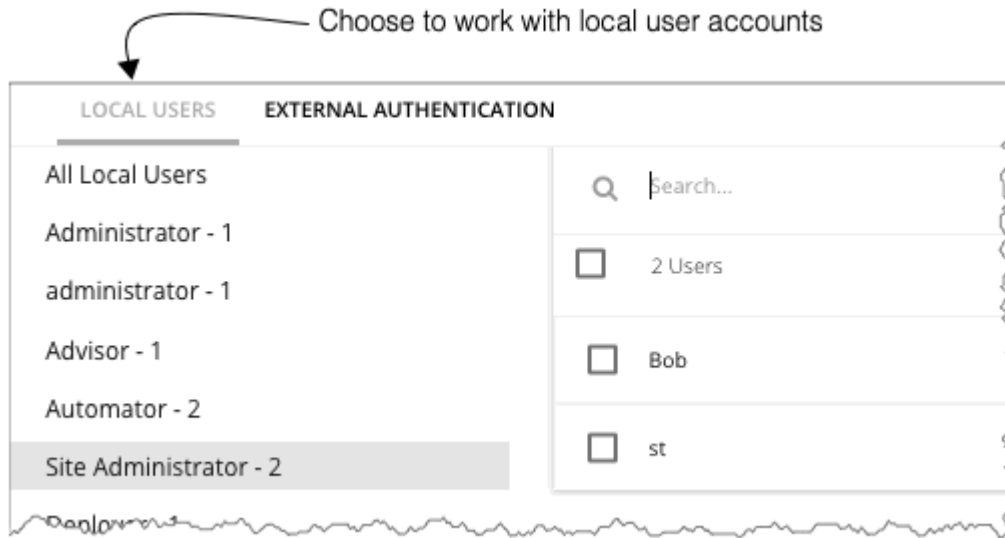
- Click to manage LOCAL USERS or EXTERNAL AUTHENTICATION
- Select an entry to delete the account
- Click a name to edit the account
- Create new user or group account
- Configure Active Directory settings

3. Filter the list of users.

This screenshot shows the 'User Management' page with filtering applied. The 'LOCAL USERS' tab is active. The user list on the left shows 'Site Administrator - 2' highlighted. The search and filter section on the right shows '2 Users' and lists 'Bob' and 'st'. Arrows point from the text 'Filter by string' to the search bar and 'Filter by role' to the highlighted 'Site Administrator - 2' entry.

To work with a long list of users, you can filter by role (for example, only show administrator or only show observer users). You can also type a string in the **Search** field to filter the list, and you can sort the list by name.

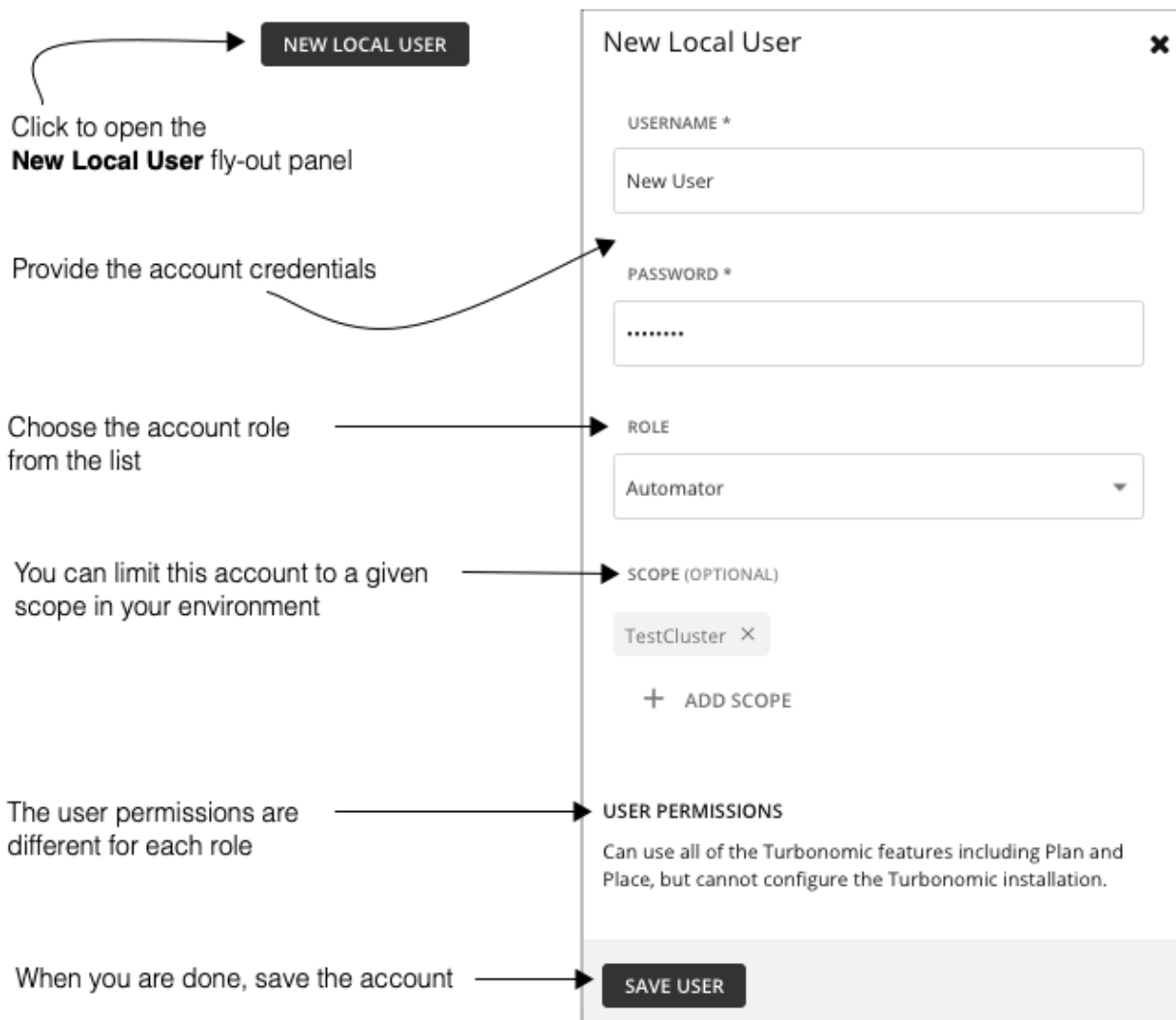
4. Work with Local user accounts.



Turbonomic stores local accounts and their credentials on the Turbonomic platform. Local authentication is for individual users, only.

When you choose **LOCAL USERS**, Turbonomic displays a list of all the local user accounts you have configured for this installation.

5. Create or edit a local user account.



To add a new local user, click **NEW LOCAL USER**. To edit an existing account, click the account name in the list. To configure a local account, specify:

- Authentication:
  - Provide the username and password. Turbonomic stores these credentials on the local server.
- Authorization – User Role:
  - Administrator
    - Users with this role can use all Turbonomic features and modify settings to configure the Turbonomic installation. For Turbonomic instances hosted in the public cloud, this role is limited to the Turbonomic representative that manages the instances.
  - Site Administrator
    - Users with this role can use all Turbonomic features and modify site-specific settings to configure the Turbonomic installation. Users can also administer Groups, Policies, Templates, Billing/Costs, and Target Configuration, but not Email, Licenses, Updates, and Maintenance. Users can create other user accounts, except accounts with the Administrator role.
  - Automator
    - Users with this role can use all Turbonomic features (including Plan, Park, and Place), but cannot configure the Turbonomic installation or create policies.
  - Deployer



Users with this role can view all Turbonomic charts and data, use Place to reserve workloads, and create placement policies and templates. However, users cannot run plans or execute any recommended actions.

- Advisor

Users with this role can view all Turbonomic charts and data, and run plans. However, users cannot use Place to reserve workloads, create policies, or execute any recommended actions.

- Observer

Users with this role can view the environment, including the Home Page and Dashboards. Users can also use Search to set a scope to the session. For scope, only VM groups and Resource Groups are supported.

- Operational Observer

Users with this role can view the environment, including the Home Page, Dashboards, Groups, and Policies. Users can also use Search to set a scope to the session.

- Shared Advisor

Users with this role are scoped users. They can view the Home Page and Dashboards, but only see VMs and Applications. Users cannot execute Turbonomic actions.

- Shared Observer

Users with this role are scoped users. They can view the Home Page and custom Dashboards, but only see VMs and Applications. Users cannot see Executive Dashboards or execute Turbonomic actions. This is the most restricted user.

- Report Editor

Users with this role can create, edit, and delete reports. Due to limits to the reporting license, only one user per instance is allowed to have this role (by default, the local **administrator** user). To assign this role to another user, you must first remove it from the current user. Be sure that the new user is *not* a scoped user.

- Authorization – Scope (optional)

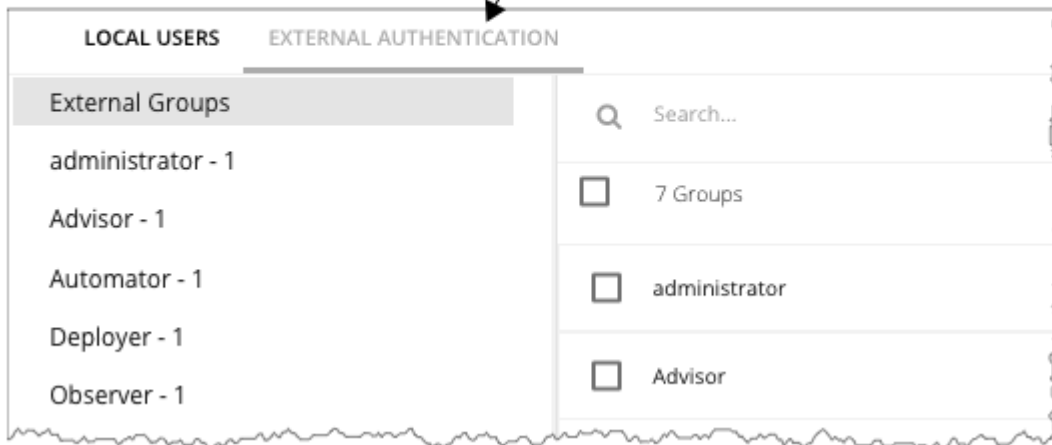
The scope limits what the user can monitor. For example, you can scope to a group that contains only the physical machines that support this user's VMs or applications. Click **ADD SCOPE** and choose which groups or clusters this user can see.

**Note:**

Under most circumstances, a scoped user cannot see actions for entities that are outside of the configured scope. However, when zooming in to Host entities, the user can see actions for storage that is outside of the user's scope if the hosts use that storage.

6. Work with EXTERNAL AUTHENTICATION to set up SSO or AD accounts.

Choose to work with externally authenticated users or groups



For External Authentication, you configure Turbonomic to use SSO or AD services to manage the credentials and authentication of users. You can create external accounts to authorize user groups or individual users.

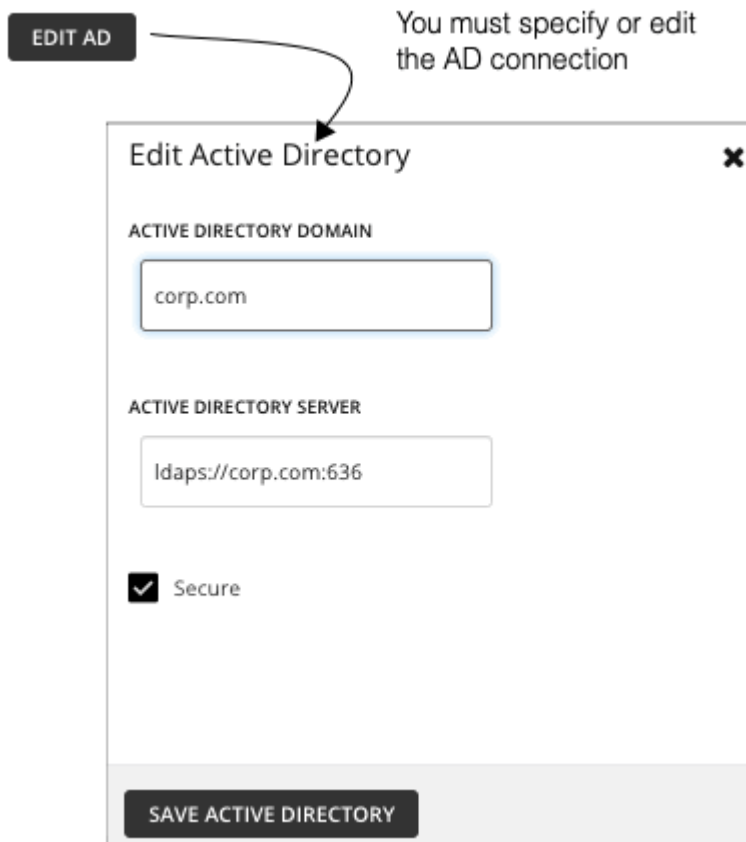
**Note:**

If a user is a member of multiple groups, then Turbonomic logs the user on via the first SSO or AD that successfully authenticates the user. Also note that Turbonomic does not support nested AD groups – AD logins must be for users in a top-level group.

To enable SSO, you must configure access to the given IdP. For information about configuring SSO, see [Single Sign-On Authentication](#).

To enable AD you must specify either an AD domain, an AD server, or both. Turbonomic uses this connection for all AD users.

7. Enable AD authentication.



To enable AD, click **CONNECT TO AD** and configure:

- Active Directory Domain – To authenticate AD groups, specify a domain so that AD can find a given user via the User Principal Name (UPN). If you specify a domain, but not a server, authentication uses any AD server from that domain.
- Active Directory Server – To disable AD groups, specify a server but do not specify a domain. If you specify a domain and a server, authentication will use that server, and will also support groups.

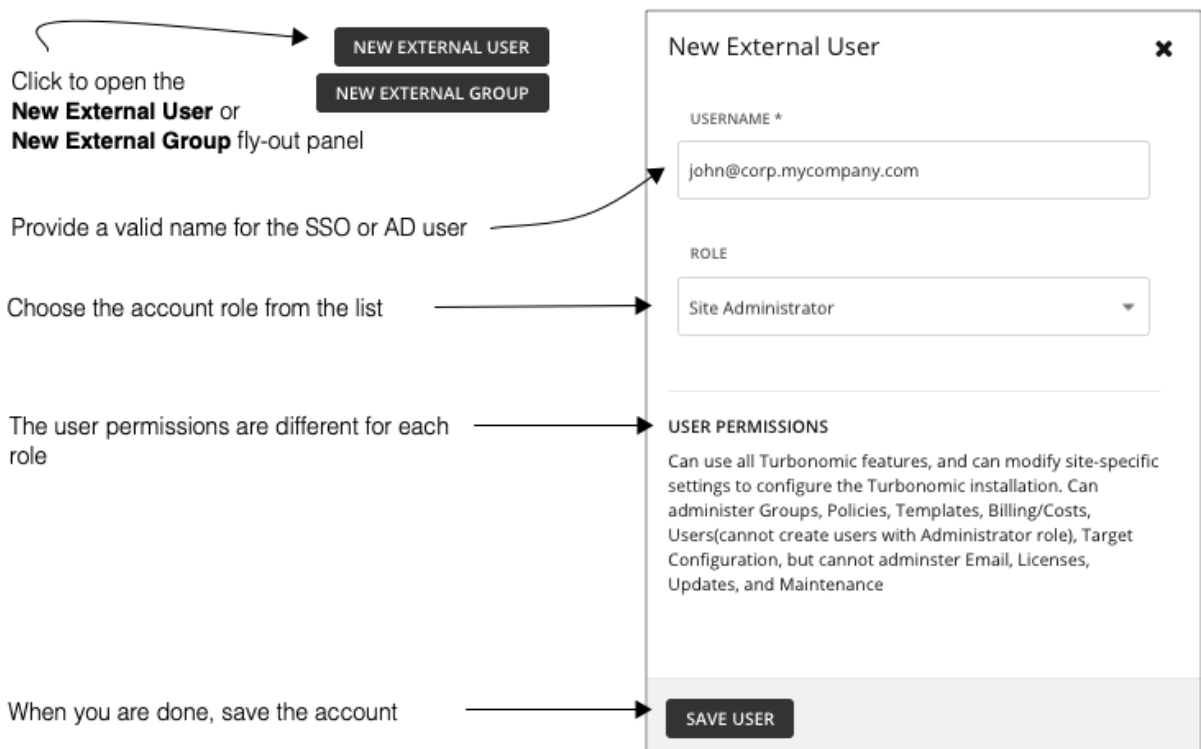
When you configure an AD server, by default Turbonomic assumes the AD server port to be 389 or 636. To specify a custom port for the AD server, add the port number to the AD server IP address. For example, `10.10.10.123:444` sets port 444.

- Secure – Use a secure connection when communicating with AD servers. Note that the AD domain must be configured to use LDAPS, and you must have imported a certificate into the Turbonomic server.

Turbonomic can support LDAP channel binding and LDAP signing. To support these Active Directory features, you must configure secure access.

For more information, see [Enforcing Secure Access](#).

8. Create or edit an SSO or AD account.



This account can be for a user group or for a single user. To add a new account, click **NEW EXTERNAL GROUP** or **NEW EXTERNAL USER**. To edit an existing account, click the account name. To configure an external account, specify:

- Authentication:

Provide the group or user name for this account. The name you provide must meet certain requirements, depending on the type of account you are creating:

- **External Group - SSO**

Provide a name that matches a group the IdP manages.

- **External Group - AD**

The group name must match a group that is accessible from the domain and servers that you configured in **EDIT AD**.

- **External User - SSO**

Provide a user name that matches a user managed by the IdP.

- **External User - AD**

The username must be a valid User Principal Name (UPN). For example, john@corp.mycompany.com.

- Authorization – User Role:

- Administrator

Users with this role can use all Turbonomic features and modify settings to configure the Turbonomic installation. For Turbonomic instances hosted in the public cloud, this role is limited to the Turbonomic representative that manages the instances.

- Site Administrator

Users with this role can use all Turbonomic features and modify site-specific settings to configure the Turbonomic installation. Users can also administer Groups, Policies, Templates, Billing/Costs, and Target Configuration, but not Email, Licenses, Updates, and Maintenance. Users can create other user accounts, except accounts with the Administrator role.

- Automator
 

Users with this role can use all Turbonomic features (including Plan, Park, and Place), but cannot configure the Turbonomic installation or create policies.
- Deployer
 

Users with this role can view all Turbonomic charts and data, use Place to reserve workloads, and create placement policies and templates. However, users cannot run plans or execute any recommended actions.
- Advisor
 

Users with this role can view all Turbonomic charts and data, and run plans. However, users cannot use Place to reserve workloads, create policies, or execute any recommended actions.
- Observer
 

Users with this role can view the environment, including the Home Page and Dashboards. Users can also use Search to set a scope to the session. For scope, only VM groups and Resource Groups are supported.
- Operational Observer
 

Users with this role can view the environment, including the Home Page, Dashboards, Groups, and Policies. Users can also use Search to set a scope to the session.
- Shared Advisor
 

Users with this role are scoped users. They can view the Home Page and Dashboards, but only see VMs and Applications. Users cannot execute Turbonomic actions.
- Shared Observer
 

Users with this role are scoped users. They can view the Home Page and custom Dashboards, but only see VMs and Applications. Users cannot see Executive Dashboards or execute Turbonomic actions. This is the most restricted user.
- Report Editor
 

Users with this role can create, edit, and delete reports. Due to limits to the reporting license, only one user per instance is allowed to have this role (by default, the local **administrator** user). To assign this role to another user, you must first remove it from the current user. Be sure that the new user is *not* a scoped user.
- Authorization – Scope (optional)
 

The scope limits what members of this group can monitor. For example, you can scope for access to only the hosts that support this group's VMs or applications. Click **DEFINE SCOPE** and choose which entities this members of this group can see.

## Configuring a Group for SSO Authentication

To use SSO authentication in Turbonomic, you should configure user groups on the IdP. The IdP can authenticate the group members, and then Turbonomic can assign the user role and scope according to that group's authentication. To manage personnel changes, you only need to manage the membership in the IdP group. For example, if a user leaves your organization, you only need to remove the member from the group on the IdP. Because authorization on Turbonomic is by group, that user will not have any authorization settings stored on the Turbonomic server.

### Important:

Before you enable SSO for your Turbonomic installation, *you must configure at least one SSO user with Turbonomic administrator privileges*. If you do not, then once you enable SSO you will not be able to configure any SSO users in Turbonomic. To authorize an SSO user as an administrator, use **EXTERNAL AUTHENTICATION** to do one of the following:

- Configure a single SSO user with administrator authorization.

Add an external user. The username must match an account that is managed by the IdP.

- Configure an SSO user group with administrator authorization.

Add an external group. The group name must match a user group on the IdP, and that group must have at least one member.

For more information about configuring SSO authentication, see [Single Sign-On Authentication](#).

## Specifying a Group in the SAML Response

To support SSO, Turbonomic recognizes IdP responses that comply with SAML 2.0. To create user groups, for each user response you include an attribute named group, and give the group name as the attribute value. For example, assuming the following users, setting the group attribute for each user assigns that user to the appropriate group.

| Users:                                                                                                                                             | Group Attribute:                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|
| <ul style="list-style-type: none"> <li>• George</li> <li>• Paul</li> <li>• John</li> <li>• Ringo</li> </ul>                                        | Attribute Name=group, AttributeValue=Beatles  |
| <ul style="list-style-type: none"> <li>• Smokey</li> <li>• Pete</li> <li>• Ronnie</li> <li>• Claudette</li> <li>• Bobby</li> <li>• Marv</li> </ul> | Attribute Name=group, AttributeValue=Miracles |

As you specify the user response, to add the user to a group you include a group attribute. For example, to add a user to a group named turbo\_admin\_group, you would include the following attribute in that user's SAML response:

```
<saml2:Attribute
 Name="group"
 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
 <saml2:AttributeValue
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xs:string">
 turbo_admin_group
 </saml2:AttributeValue>
</saml2:Attribute>
```

## Setting Group Authorization in Turbonomic

To set an account role and scope to a user group, you must use the group name that you specify as the value in the given SAML group attribute. In the above example, the group value is turbo\_admin\_group. To set authorization for that group:

1. Open the User Management page to EXTERNAL AUTHENTICATION.
  - Navigate to **Settings > User Management**, and display the **EXTERNAL AUTHENTICATION** view.
2. Create a new External Group
  - Click **NEW EXTERNAL GROUP**.
3. Provide the group name.

Be sure to use the name that you specify in the group attribute of the SAML response. For the above example, use the name turbo\_admin\_group.

#### 4. Specify the group's authorization

For the above example, since this is turbo\_admin\_group, you should set the **ADMINISTRATOR** role, and you should not set any scope (grant full access to the environment).

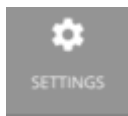
After you configure this group in Turbonomic, then any member of turbo\_admin\_group that the IdP returns will have full administrator privileges on your Turbonomic installation.

## Maintenance Options

The Maintenance Options Page provides tools to set logging levels and to export data for technical support, and import diagnostic files from Technical Support. Many of these tools are for advanced users. You should contact IBM technical support before you use them.

To execute these actions, navigate to the Maintenance Options page.

#### 1. Navigate to the Settings Page.



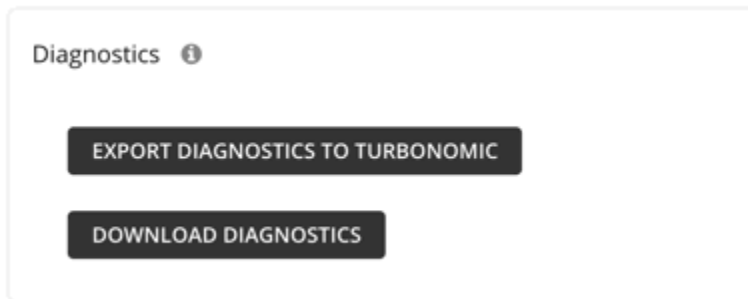
Click to navigate to the Settings Page.

#### 2. Choose Maintenance Options.



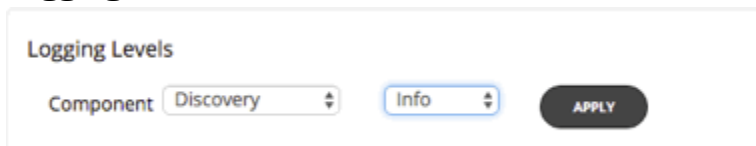
Maintenance Options

## Diagnostics



If you are experiencing problems with Turbonomic, your support engineer might request that you export diagnostic data. You can export the data and then send it to the support engineer as requested.

## Logging Levels



You can set the level of logging for different components of the Turbonomic platform. You should be aware that setting more verbose logging levels increases the disk space required to store the log files. You normally change these settings only while you're working with a Turbonomic support engineer.

## Feedback and Diagnostics

**Feedback and Diagnostics**

Allow Turbonomic to collect anonymized usage data and non-confidential environment statistics to improve your experience. We never collect usernames, passwords, server names, IP addresses, page content, or other information of this kind.

Share anonymized usage data and environment statistics

**APPLY SETTING AND REFRESH**

Turbonomic relies on feedback from customers to continuously improve the product. One way that we receive feedback is by collecting anonymized and non-confidential data as users go about using the product. Your participation in this initiative is voluntary (and you can opt out at any time), but it does help Turbonomic deliver a better user experience for customers.

You can enable this feature to participate. Note that you must have the **Administrator** role to enable or disable this feature. If enabled, Turbonomic collects browser event data for *all* users logged in to the product, and reviews the data on an ongoing basis to make improvements.

For example, when a product user opens the Plan page in the user interface, Turbonomic records the loading of that page as a browser event and then collects non-confidential data, such as how long it took to load the page and on which browser. Turbonomic will never collect sensitive data, such as the user's IP address or location. If page loading is consistently slow over time and across users, Turbonomic can introduce measures to improve performance.

When you install the product for the first time, you are given the option to enable this feature. For product upgrades, Turbonomic keeps the previous version's setting. You can always change the setting from the Maintenance Options page.

### Note:

To learn more about Turbonomic's privacy policy, visit <https://www.turbonomic.com/privacy-policy>.

## HTTP Proxy

**HTTP Proxy**

**APPLY CONFIGURATION**  Enable Proxy

If your environment requires an HTTP proxy for Turbonomic to access the web, provide the credentials here.



## Data Retention

### Data Retention

|                                  |                                 |                                 |
|----------------------------------|---------------------------------|---------------------------------|
| SAVED AUDIT-LOG ENTRIES          | DAILY SAVED STATISTICS          | HOURLY SAVED STATISTICS         |
| <input type="text" value="365"/> | <input type="text" value="60"/> | <input type="text" value="72"/> |
| Days                             | Days                            | Hours                           |
| MONTHLY SAVED STATISTICS         | SAVED PLANS                     | SAVED REPORTS                   |
| <input type="text" value="24"/>  | <input type="text" value="14"/> | <input type="text" value="30"/> |
| Months                           | Days                            | Days                            |

Turbonomic gathers metrics from your environment to provide historical reports. To optimize data storage, it consolidates the data into three groups - Hourly, Daily, and Monthly. Daily statistics consolidate Hourly data, and Monthly statistics consolidate Daily data. Turbonomic also saves plans, reports, and audit log entries.

You can always modify the default values to meet your requirements. Remember that the longer the retention period, the more storage is required.

## The Updates Page

Use the Updates page to get information about your Turbonomic version. Note that this page is not available in SaaS instances of Turbonomic.

The **ABOUT** button shows the current version and build of your Turbonomic installation. It also lists the platform components by name and version.

### Note:

For complete update instructions, see [Updating to a New Version](#).

To navigate to the Updates page:

1. Navigate to the Settings Page.

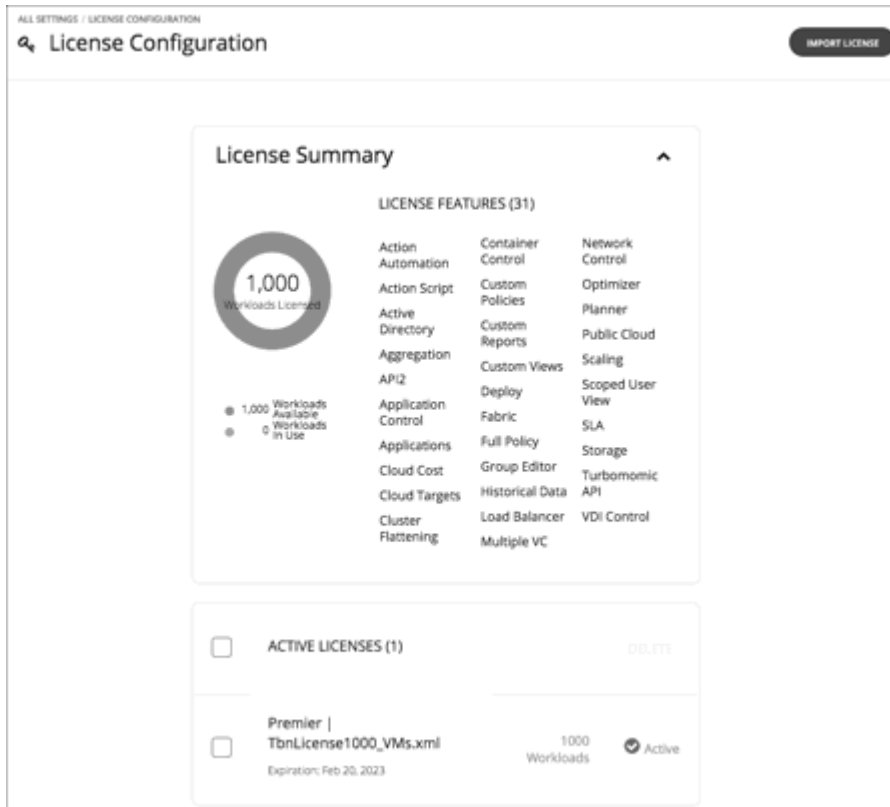


Click to navigate to the Settings Page.

2. Choose Updates.



## License Configuration



To activate the full range of Turbonomic features, you must purchase the appropriate license. When you purchase the license, IBM sends an e-mail message with instructions on how to obtain the license key.

### Note:

If you purchased Turbonomic SaaS, IBM sends an e-mail message with instructions on how to access the Turbonomic UI, including the URL for your instance and your credentials. You will get a customer-specific license key for the number of licenses and duration of the subscription you purchased.

Turbonomic bases its licensing on the number of workloads a given license supports. If your environment exceeds that count of workloads, then you are no longer in compliance with your Turbonomic requirements, and some features will be restricted. For example, you will not be able to configure new targets. You should upgrade your license as soon as possible.

To upgrade your license to support a larger environment, contact your Turbonomic representative to get the correct license, and to make sure you know how to install it correctly.

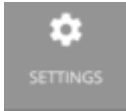
A product license enables specific features as well as a specific number of workloads that you can manage. You can add additional licenses to Turbonomic as a way to increase the number of workloads your installation can manage. Note that as you add more licenses, they must all support the same feature set.

The License Configuration page shows you:

- The number of workloads you can manage under this license
- How many workloads are currently in use
- The set of features this license enables
- A list of each license and its status

To navigate to the License Configuration page:

1. Navigate to the Settings Page.



2. Choose License.



To activate a license or to update your current license:

1. Obtain your license.

IBM sends an e-mail message with instructions on how to obtain the license key. Save the license file on your local machine so you can upload it to your Turbonomic installation.

2. Apply the license to your Turbonomic installation.

First click **IMPORT LICENSE**. Then browse to the license file that you saved and open it. Or you can drag the file into the **Enter License** fly-out.

After you have uploaded the file, click **SAVE**.

After you have activated your license, you can then add more licenses to increase your workload coverage, or you can license a higher feature set.

**Note:**

This only applies only to legacy Turbonomic customers. As you apply new licenses to Turbonomic, you must be sure that they are for the same edition or feature set. If you try to apply an incompatible license file, Turbonomic displays an *Invalid Feature Set* error. To apply the new license you must either delete your current license so you can install the new feature set, or you must obtain a different license file that matches your current feature set.

After you install a new license, you should clear your browser cache and reload the Turbonomic user interface.

To increase your licensed workload coverage:

1. Obtain your additional license.

Note that your additional licenses must match the feature set of your current license.

2. Apply the license to your Turbonomic installation.

To upgrade your license to a higher feature set:

1. Obtain your new license for the new features.

You should obtain a license that supports at least the same number of workloads as your current license.

2. Delete your current license from Turbonomic.

On the license page, select all the licenses that you currently have installed, then click **DELETE**.

3. Apply the license to your Turbonomic installation.

## Email Settings

Configure email settings to enable email communication from Turbonomic.

1. Navigate to the Settings Page.



Click to navigate to the Settings Page. From there, you can perform a variety of Turbonomic configuration tasks.

## 2. Navigate to the Email Settings Page.

From here, you can configure:

- SMTP Settings
- General Email Settings

### SMTP Settings

SMTP Settings ⓘ

|                                      |                                 |
|--------------------------------------|---------------------------------|
| SMTP SERVER                          | SMTP PORT                       |
| <input type="text" value="0.0.0.0"/> | <input type="text" value="25"/> |
| USERNAME                             | PASSWORD                        |
| <input type="text"/>                 | <input type="text"/>            |

ENCRYPTION

None

Ssl

Tls

The SMTP Settings fields identify the mail relay server you use on your network to enable email communication from Turbonomic.

If the server requires authentication, provide the username and password here. You can also choose the following encryption options for notifications:

- None
- Ssl
- Tls

### General Email Settings

General email settings

FROM ADDRESS

Use this setting to specify the return address (the FROM address) for emails that Turbonomic generates and sends.

---

## Chapter 9. API Reference

This guide will help you to use the Turbonomic REST API as you script interactions with the Turbonomic software and develop integrations between Turbonomic and other software applications.

The REST API exposes Turbonomic data and processing to remote access via HTTP GET, POST, PUSH, and DELETE methods.

The Turbonomic user interface is a client to this API, and every feature exposed in the user interface is the product of API calls. To understand the API, you should have a working knowledge of the Turbonomic user interface, the features it exposes, the data it uses, and the use cases that it supports. In most cases, if you want to script some Turbonomic behavior, you should be able to execute most of that behavior via the user interface. Stepping through these use cases in that context will help you plan and implement your scripts.

To get the most out of the REST API, you should understand how Turbonomic organizes its underlying data, and how the various REST resources map to that organization. This introductory section includes discussions of the data that underlies:

- Markets and how they represent your environment (see [Markets](#))
- Plans and an overview of how to work with them (see [Plans](#))
- Workload placement policies and how they are represented (see [Workload Placement Policies](#))
- Reservations for deployment, and how to set them up (see [Reservations](#))

### Note:

This guide is not a comprehensive reference to every possible setting and data payload in the REST API. Many endpoint descriptions include samples of API data objects, but they cannot document every possible combination of inputs. For a complete reference to the API, use this guide in conjunction with the Turbonomic REST Swagger-UI documentation. For more information, see [Turbonomic REST API Swagger Documentation](#).

---

## API Guide Revision History

A revision of the API Guide is available for each major release, if the API undergoes significant change between major releases, or to correct significant errors in the API Guide. Each revision of the API Guide includes the following changes:

- Quarterly (major) release - the new revision for the major release, includes fixed issues and improvements to the API
- Bi-weekly (point) release - deprecated features are removed from the API or fully implemented as outlined in the API Deprecation History

For a list of upcoming deprecations, see [API Deprecation Contract](#).

| Version Family          | Releases                                      |
|-------------------------|-----------------------------------------------|
| 8.9                     | <a href="#">8.9.1</a>   <a href="#">8.9.0</a> |
| 8.8                     | <a href="#">8.8.1</a>   <a href="#">8.8.0</a> |
| 8.7                     | <a href="#">8.7.1</a>   <a href="#">8.7.0</a> |
| 8.6                     | <a href="#">8.6.1</a>   <a href="#">8.6.0</a> |
| Versions older than 8.6 | <a href="#">Archives</a>                      |

## Getting Started with the Turbonomic REST API

---

The easiest way to see results with the API is to try calls in the Swagger UI. To use the Swagger UI, navigate to:

```
https://<Your_Turbonomic_IP>/swagger/external/index.html
```

You can use the Swagger UI to navigate to different endpoints and try out their methods. For example, if you navigate to the Users endpoint you can get a listing of all the users that are specified for your Turbonomic installation. For more information, see [Turbonomic REST API Swagger Documentation](#).

Ultimately, you will want to implement scripts that use the API to get data, execute actions, or integrate Turbonomic with other processes. As you work with the API, you need to know about:

- Authentication
- URI Structure
- Response Format
- Time in the Turbonomic API

### Authentication

To use the API, you must have a valid user account on the Turbonomic instance. Also note that accounts can have different roles. The API will only execute commands that are valid for your user role. For example, to execute Turbonomic recommended actions, your account must have a role of either `administrator`, `deployer`, or `automator`.

To make API calls, you request an authentication token and pass it with each call to the Turbonomic API. The token request returns a cookie for your authentication. A common way to use this token is to store the cookie locally, and pass it with your API calls.

**Example:** `curl -s -k -c /tmp/cookies -H 'accept: application/json' 'https://localhost/api/v3/login?hateoas=true' -d 'username=administrator&password=password'`

Then, each request must use the `-b cookie-filename` parameter to use the session cookie delivered by the login request.

Another approach is to get the authentication header and parse out the authentication cookie. Then you can create a header for each API request that includes the cookie. For example, assume you store the value in a variable named `token`. You could use it like this:

```
headers = {'cookie': token}
r = requests.get('https://10.10.123.456/api/v3/targets/specs', headers=headers, verify=False,
stream=True)
```

**Note:** For a more complete explanation, see the [Authentication Recipe](#) in the API Cookbook.

### URI Structure

To use the Turbonomic REST API, your client will make HTTP requests to specific REST resources. The Turbonomic REST API supports the standard HTTP methods:

- GET  
Get lists of entities or data objects, get individual items.
- POST  
Create new objects in the Turbonomic environment, or specify filters for certain queries.
- PUT  
Incrementally modify existing entities or objects.
- DELETE

Delete entities or objects.

The base URI structure for a Turbonomic REST API resource is:

```
https://<Your_Turbonomic_IP>/api/<API_version>/<resource_name>
```

For example, to list the users in your installation:

```
https://111.222.33.44/api/v3/users
```

## Response Format

The Turbonomic REST API returns data as JSON objects. Turbonomic refers to these objects as Data Transfer Objects or DTOs. A DTO is an array of key-value pairs that describe the data you have requested, or the data that is the result of executing a POST or PUT. For example, if you GET the users defined for an installation of Turbonomic, the API returns a DTO similar to this:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/users/_4T_7kwY-Ed-WUKbEYSVIDw"
 }
],
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "displayName": "Administrator User",
 "username": "administrator",
 "roleUuid": "_4UAioQY-Ed-WUKbEYSVIDw",
 "roleName": "administrator",
 "loginProvider": "Local",
 "type": "DedicatedCustomer",
 "showSharedUserSC": false
 }
]
```

In this case, the DTO is an array of one object. In other words, there is only one user account defined for this installation of Turbonomic. The user object begins with a `links` array that gives the URL to this user account. It then follows with properties to describe the given user account.

Note that in most cases to execute a PUT or POST, you will pass the parameters to create the object via a DTO. These DTOs are similar to the associated response DTOs, but they are not identical. For this example of a user account, the response DTO does not include the user account password, but the DTO to create an account must include the password.

You can try out different REST methods in the Swagger UI to see typical response DTOs.

## Time in the Turbonomic API

The Turbonomic REST API contains requests that require a start time and/or an end time to create or filter information. Time may be entered in three different formats:

- ISO 8601 Date and Time Format

YYYY-MM-DDTHH:MM:SS. For example, 2018-10-07T12:38:17

- Epoch Time

Epoch Time is represented as the number of seconds that have elapsed since midnight UTC, January 1, 1970. For example, 1514764800 corresponds to January 1, 2018 12:00:00 AM UTC.

- Relative time

Relative time is represented as the time relative to when the call is executed. For example, a start time of `-1w` and an end time of `-1d` indicates that the results should include entries from a week before the

call is executed, to the day before the call is executed. Relative units are case-sensitive. Turbonomic supports the following relative units:

- m – minutes
- h – hours
- d – days
- w – weeks
- M – months
- y – years

## Pagination in the Turbonomic API

Some API calls return very large datasets. Turbonomic recommends using the pagination features at all times. This is particularly important for larger environments managed by Turbonomic. Via scopes and filters for individual requests, data may be preliminarily filtered. Using the `limit` and `x-next-cursor` parameters, the data can be returned in manageable chunks.

For example:

```
https://10.10.10.10/api/v3/markets/214075923753936/entities/stats?
limit=5&ascending=true
```

The request above is to obtain statistics for all entities in a specific market. `limit=5` indicates that each page of returned data should have 5 results.

The headers in the returned data will now include `x-next-cursor: 5`. In order to retrieve the next page of results, use the cursor in the next request:

```
https://10.10.10.10/api/v3/markets/214075923753936/entities/stats?
cursor=5&limit=5&ascending=true
```

When the final page is reached, the `x-next-cursor` will be empty.

If a request features the `orderBy`, `limit`, or `cursor` parameters, using one of those parameters will return paginated results.

## Turbonomic REST API Swagger Documentation

---

You can try out different REST methods in the Swagger documentation to see typical response DTOs.

To access the Turbonomic Swagger documentation, navigate to:

```
https://<Your_Turbonomic_IP>/swagger/external/index.html
```

This documentation uses a standard Swagger-UI application that delivers documentation for all the methods on the different endpoints in the API. This documentation includes a description of the call, a list of parameters to pass, and descriptions of the JSON payloads you POST or PUSH, as well as the payloads you get in the response. If you view the MODEL of a JSON payload, you can navigate the JSON structure to see descriptions of the different object fields.

To access the API, the Swagger-UI requires authentication. If you have already logged into Turbonomic, your browser session should have a current authentication cookie. The Swagger-UI uses this same cookie for its API authentication.

For each method, the Swagger doc includes a **Try it out** feature where you can set up a call through the Swagger-UI, and then execute it. This generates the following information as a response:

- A curl request for your call

The curl commands generated by Swagger may need some modification for use in your environment (for example, escaping quotes or adding the `-k` modifier in order to use HTTP URLs).

- A standard HTTP request for your call
- The server response code



- The response body – The JSON data that the API call returns
- Response headers

**Note:** If you use the SwaggerUI to try out API calls, and the API call returns an error (for example, your call uses incorrect syntax), under some circumstances SwaggerUI does not display the error, and the field for the returned data is empty.

## Turbonomic Markets

---

Turbonomic uses market-based analysis to perform workload management. To do this, it constructs a model of your environment, representing each entity as a buyer and seller in a market. You can access this model via a named market resource.

At any time, your Turbonomic appliance can have a number of markets in memory. To get a list of the current markets in your appliance, execute the following URL:

GET: /rest/markets

This listing returns the main real-time market, plus any plan markets that are in memory at the time of the call. Note that you can pass group UUIDs to set a scope for the call. In that case, the call returns only markets for plans that include the passed groups in their scope.

### The Real-Time Market

The real-time market performs analysis and workload management on your environment. You can use the real-time market to access entities and get current or historical data about them.

The real-time market displayName is Market. This market should always be in the RUNNING state. Under no circumstances should you use the API to stop this market.

Turbonomic performs discovery to populate the topology it manages (the collection of entities in the real-time market). For this reason, you should be careful not to delete entities from or add entities to the real-time market.

For the real-time market, you can safely execute POST, PUT, and DELETE calls to:

- Post a query filter to get filtered lists of actions, notifications, and stats for the real-time market
- Post to create placement policies in the real-time market
- Delete to remove placement policies from the real-time market
- Put to edit placement policies on the real-time market
- Post a scenario to the real-time market to run a plan

When you run a plan scenario against the real-time market, Turbonomic creates a copy of the real-time market, and modifies the copy according to the plan scenario.

**Note:**

It is possible to execute POST or DELETE methods to modify the topology of the real-time market. However, it is highly unlikely that you would have a reason to do so. You should be aware that changes to the real-time market will affect the analysis Turbonomic performs. For example, if you remove a VM from the real-time market, Turbonomic can no longer manage its placement. The VM will still be present in your environment, but it will no longer be managed by Turbonomic. However, you can't reliably use this technique to make specific entities unavailable to Turbonomic. For the next discovery pass, Turbonomic will rediscover the deleted entity, and it will appear in the real-time market again.

### Plan Markets

Turbonomic can have markets other than the real-time market in memory. These other markets represent plans.

A plan market begins as a snapshot of the real-time market. You POST a scenario to the real-time market, and Turbonomic does two things:

- It makes a copy of the real-time market, to create a plan market. This plan market is just like the real-time market, except for any changes that were declared in the scenario. For example, if the scenario declares a scope for the plan, then the plan market only contains entities that are within that scope.
- It runs economic cycles (buy/sell cycles) against this plan market until there are no more meaningful improvements to be made. At this point the plan run is completed.

Note that once Turbonomic creates the plan market, that market stays in memory until you delete it. Also note that you can later apply a scenario to the plan market. This is how you run a plan on a plan. The logic flow is the same, and Turbonomic creates yet another plan market to run the analysis.

## Internal-Use Plans

Turbonomic regularly runs plans to generate data that it displays in the GUI — The Cluster Capacity and Project Cluster Resources dashboards both display data generated by regularly-run plans.

A plan market that is for internal use includes the substring `_BasePlan` in the `displayName`. You should not modify these plan markets for any reason.

## User-Created Plans

Users can create plans to run what-if scenarios in the environment, and the results are saved as a plan market. At any time, each user account can have multiple plan markets loaded in the Turbonomic instance.

You identify planner markets by their names. A plan name is specified as `<PlanType><userName>_<arbitraryID>`.

For example, a valid plan name is `CLOUD_MIGRATION_cud_1493140514716`. To find out which user owns this plan, you can parse out the user's name and query the API for that user's account information.

When you get a plan market, the response includes information such as:

- `uuid`: The market's identifier.
- `state`: Whether the plan succeeded or not. The state can be one of:
  - `CREATED`
  - `READY_TO_START`
  - `RUNNING`
  - `COPYING`
  - `SUCCEEDED`
  - `STOPPING`
  - `STOPPED`
  - `USER_STOPPED`
  - `DELETING`

As long as a plan market is running, Turbonomic is still calculating the plan results. If the market is stopped, then the plan has been run and you can access data from this market to see the plan results. You can make a PUT call to the market to stop a plan that is running.

For a plan market, you can execute the same POST, PUT, and DELETE methods that you would execute on the real-time market. These include:

- POST a query filter to get filtered lists of actions, notifications, and stats for the plan market
- POST to create placement policies in the plan market
- DELETE to remove placement policies from the plan market
- PUT to edit placement policies on the plan market
- POST a scenario to the plan market to run a plan on plan

You can also safely execute the following methods on a plan market:

- DELETE a plan market
- PUT to save or stop a plan market

**Note:**

It's possible to stop a plan market through the GUI or the API. Also, a plan may have been stopped before it finished its calculations. In this case, the plan results will be incomplete.

## Running Plans

---

A plan is a simulation or what-if scenario that explores the results of possible changes to your environment. To run a plan, Turbonomic creates a snapshot copy of your real-time market and applies changes to it. It then uses the Economic Scheduling Engine to perform analysis on that plan market.

Before working with plans in the API, you should be familiar with plans via the GUI. You should know what plans can accomplish, and how to specify the plan settings such as:

- Plan scope
- Changes to workload in the environment (adding or removing VMs)
- Changes to supply (adding, removing, or reconfiguring hardware)
- Changes to placement (migrating to the cloud or a different cluster)
- Changes to constraints

To create a plan, you first specify a plan scenario. This is an object that contains all the plan settings. The scenario object contains an array of changes, and each change declares a setting that you want to make.

After you create a scenario, you POST it to a market. In most cases you will post it to the real-time market. When you post the scenario to the market, Turbonomic does two things:

- It makes a copy of the market, to create a plan market. This plan market is just like the original market, except for any changes that were declared in the scenario. For example, if the scenario declares a scope for the plan, then the plan market only contains entities that are within that scope.
- It runs economic cycles (buy/sell cycles) against this plan market until there are no more meaningful improvements to be made. At this point the plan run is completed.

### Saving and Deleting Plan Markets

As you run a plan, Turbonomic performs analysis and shows the results. It also serializes and saves the plan results so users can load the results into the GUI at a later time.

**Note:**

Updates to Turbonomic can change the data format for the saved plan results. In that event, then the results will be inconsistent. After each update to Turbonomic, you should regenerate all the plans you want to save, and save them again.

To delete a plan via the API, just execute a DELETE method on the plan market.

### Creating Scenarios

The first step for a plan is to create the scenario. Note that a scenario is entirely separate from the plan. You create a scenario and then POST it to a market to create the plan. But the scenario still exists in memory, and you can apply it to a different market if you wish. For example, you can apply the same scenario to the real-time market at different times.

To create a scenario, use the POST: `/rest/scenarios/{name}` method. This method takes three parameters:

- scope

An array of group UUIDs. For a scope of multiple groups, you must specify groups of the same type.

- `projection_days`

For a projection scenario, the days to mark each projection period

- `input`

A `ScenarioApiInputDTO` that can specify all the settings for a scenario.

The `input` parameter, a `ScenarioApiInputDTO`, is an object that you can POST to the API to specify scenario settings. The API includes methods to create and delete scenarios. In most cases, you will create a scenario by POSTing or PUTing a `ScenarioApiInputDTO` with changes to declare the scenario settings you want. Note that the API includes methods you can use to PUT many of these settings into a specific scenario by passing properties directly in the URL. However, you can also make these settings with the `ScenarioApiInputDTO`. Learning to use this DTO is the most consistent way to create and modify a scenario.

Note that as you create a scenario you can POST all of its settings in the `ScenarioApiInputDTO`, or you can create an incomplete scenario, and then PUT other changes into the scenario at a different time.

## Creating a Projection Scenario

Projection plans calculate infrastructure requirements into the future, so your environment can accommodate changes to workload requirements over time. In a single plan you can specify the scope of the plan, how far into the future to project, and by what increments of time. For example, you can project the requirements of a specific datacenter one year into the future, showing how requirements change at one-month intervals.

When you create a projection scenario via the API, you provide the projection periods as an array of the days from today that you want to project. For example, 0 for today, 30 for 30 days from today, 60 for 60 days from today, and so on.

To create a scenario for a projection plan:

- Scope the scenario

A projection plan requires a scope to the scenario.

- Create a change in the plan of type `PROJECTION_PERIODS`

This change identifies the plan periods that you want, as an array of days.

```
{
 "changes": [
 {
 "type": "PROJECTION_PERIODS",
 "projectionDays": [0, 30, 60, 90]
 }
]
}
```

- For any plan changes that you want to repeat on specific projection periods, specify a `projectionDays` array for that change. Note that the items in this array must match the `projectionDays` that you specified for the `PROJECTION_PERIODS` change.

For example, assume you specified `"projectionDays": [0, 30, 60, 90]` for the initial projection:

- Valid: `"projectionDays": [0, 30, 60, 90]`
- Valid: `"projectionDays": [30, 90]`
- Not valid: `"projectionDays": [0, 35, 70, 90]`

This listing shows a plan scenario that scopes the plan market, sets up a projection for four periods, and adds one VM to the inventory for two out of four projection periods:

```
{
 "displayName": "My Projection",
 "changes": [
 {
```

```

 "type": "SCOPE",
 "scope": [
 {
 "uuid": "5678a46e9716657be88b5e1217df91436e13e4ff",
 "className": "Group",
 "entitiesCount": 2,
 "groupType": "VirtualMachine",
 }
]
 },
 {
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0,
 30,
 60,
 90
]
 },
 {
 "type": "ADDED",
 "targets": [
 {
 "uuid": "564d89f4-190b-de1b-cb28-3e2f2126ab48"
 }
],
 "value": "1",
 "projectionDays": [
 30
]
 }
]
}

```

## Placement Policies in Plans

You can enable or disable placement policies in a plan. This is useful to see how the environment would change if you had a different set of constraints. It's important to note that a placement policy exists in the context of a market. To get a placement policy, query the market that contains it. To create a placement policy for a plan, you will create the policy and add it to the plan market.

If you want to enable or disable a placement policy in a plan, you specify that as a change in the scenario. Because the placement policy exists in a market, you have to know which market you will run the plan against, and get the policy from that market.

For example, if you will run the plan against the real-time market, you would:

- Query the real-time market for the policy you want
- Give that policy's UUID in the scenario change object
- In the scenario change object — Specify to enable or disable the policy
- When the scenario is complete, POST the scenario to the real-time market

Remember that when you run a plan, Turbonomic creates a new plan market. This plan market will contain a copy of the placement policy, and will enable or disable it, depending on your setting.

## Workload Placement Policies

Workload Placement Policies affect how Turbonomic calculates placement for the workloads in your environment. Policies can enforce specific business rules to constrain placement, and they can remove boundaries to enable placement calculations across cluster or datacenter boundaries.

A placement policy exists in the context of a market — Either the real-time market or a plan market. Each market has its own set of placement policies. Even if a plan market is an exact copy of the real-time market, the plan market has its own placement policies, each with its own UUID.

Before you work with placement policies via the API, you should understand how they work in the user interface. You should understand the types of placement policies you can create, and the effect of each on market analysis.

The API supports the following types of policies:

- **AT\_MOST\_N**: Only the given number of consumers can run on a single member of the providers group. This is set in the `capacity` property of the policy object.
- **BIND\_TO\_GROUP**: The consumers can only run on members of the provider group.
- **BIND\_TO\_COMPLEMENTARY\_GROUP**: The consumers cannot run on any members of the provider group.
- **MUST\_RUN\_TOGETHER**: These consumers must run on the same provider entity.
- **AT\_MOST\_N\_BOUND**: Only the given number of consumers can run on a single member of the providers group, AND The consumers can only run on members of the provider group.
- **MERGE**: Remove cluster boundaries for the specified clusters.
- **BIND\_TO\_GROUP\_AND\_LICENSE**: Create a license group.

When you get a policy, the returned object describes the policy type, as well as the consumer and provider groups. When you create a policy, you do not have to provide the full data. You provide:

- `buyerUuid`: The group of consumers for this policy.
- `sellerUuid`: The provider group for this policy.
- `type`: The policy type.
- `policyName`: A display name for this policy.
- `enabled`: Whether to enable the policy in the market. Can be `true` or `false`.
- `capacity`: For an **AT\_MOST\_N** or **AT\_MOST\_N\_BOUND** policy, the number of consumers to allow on a provider entity.
- `mergeType`: For a **MERGE** policy, the type of clusters to merge. Can be one of `Cluster`, `StorageCluster`, `DesktopPools`, or `DataCenter`.
- `mergeUuids`: The groups that you want to merge. The group type must match the `mergeType`.

For example, to create a DON'T PLACE policy, post the following input Dto to the market:

```
{
 "buyerUuid": "f82dbbc2b3366052f3bc1ac8a68c9c06b0eb182a",
 "enabled": false,
 "policyName": "PolicyFromApi",
 "sellerUuid": "4a2f5f132ae690af147ccfd6ea9839e79da3db79",
 "type": "BIND_TO_COMPLEMENTARY_GROUP"
}
```

To edit a placement policy, PUT an input Dto to the given policy. Specify the changes you want in the input Dto.

## Calculating Reservations and Workload Placement

---

Turbonomic includes the capability to reserve resources for VMs you plan to deploy in the future. Turbonomic calculates optimal placement for these VMs and then reserves the host and storage resources that they need.

To reserve VMs, you will choose a VM template, specify any placement constraints, set how many instances to reserve, and then indicate whether to reserve now or in the future. Because reserved VMs do not yet exist, they do not participate in the real-time market. VM templates specify the resource requirements for each reserved VM, including:

- Compute and storage resources allocated to each VM
- Consumed factor. This is the percentage of allocated CPU, memory, or storage that the reserved VM will utilize.

Note that before using the API to work with reservations, you should understand how reservations work from the user interface.

With the API, you can perform the following:

- Get list of reservations
- See the current placement for the reservation workloads
- Create reservations
- Delete a current reservation

## Getting Reservation Information

To get a list of current reservations, execute `/rest/reservations`. This returns a list of all active reservations. If you know the UUID of the reservation you want, you can pass it to get data for just that reservation.

Each reservation object includes:

- Display name
- Status — can be:
  - DEPLOYING — Turbonomic is deploying the workload
  - DEPLOY\_SUCCEEDED — The workload was successfully deployed
  - IN\_PROGRESS — Placement calculation is in progress
  - PLACEMENT\_SUCCEEDED — For a new reservation, the environment has sufficient resources to place the workload; if you specified a reserve date, this will be an active reservation
  - PLACEMENT\_FAILED — For a newly created reservation, the environment doesn't have resources to place the workload; if you specified a reserve date, this will be an unfulfilled reservation
  - RETRYING — Turbonomic is trying to place the workload of an unfulfilled reservation
- Time the reservation was created, time to deploy, and time it will expire
- A description of the reserved workload
- The deployment profile that identifies the physical files that will be copied to deploy the workload, as well as optional placement limitations.
- Statistics for the compute and storage resources the reservation sets aside

For example, this listing shows a reservation for one VM:

```
{
 "uuid": "_kWZHIDmEeePgeXuo0RRbw",
 "displayName": "MyReservation",
 "count": 1,
 "status": "PLACEMENT_SUCCEEDED",
 "reserveDateTime": "Thu May 04 16:27:29 UTC 2017",
 "expireDateTime": "Thu Aug 31 16:27:29 UTC 2017",
 "deployDateTime": "Thu Aug 31 16:27:29 UTC 2017",
 "reserveCount": 1,
 "demandEntities": [
 {
 "uuid": "_kWgb7TDmEeePgeXuo0RRbw",
 "displayName": "MyReservation_C0",
 "className": "VirtualMachine",
 "template": {
 "uuid": "T423f548d-cadc-e525-6df4-1f90724cf696",
 "displayName": "vsphere-dc3.dev.mycorp.com::TMP-SUSE64",
 "className": "VirtualMachineProfile"
 },
 "deploymentProfile": {
 "uuid": "_gHJ0ICXxEeePgeXuo0RRbw",
 "displayName": "DEP-SUSE64",
 "className": "ServiceCatalogItem"
 }
 }
],
 "placements": {
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCpu",
 "value": 1
 }
]
 }
]
 }
}
```

```

 "name": "cpuSpeed",
 "value": 2603
 },
 {
 "name": "cpuConsumedFactor",
 "value": 0.5
 },
 {
 "name": "memorySize",
 "value": 2097152
 },
 {
 "name": "memoryConsumedFactor",
 "value": 0.75
 },
 {
 "name": "ioThroughput",
 "value": 0
 },
 {
 "name": "networkThroughput",
 "value": 0
 }
],
 "provider": {
 "uuid": "Virtual_ESX_42381da5-12fa-1e82-2f1c-887419380d43",
 "displayName": "hp-esx21.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 },
 "storageResources": [
 {
 "stats": [
 {
 "name": "diskSize",
 "value": 18432.363
 },
 {
 "name": "diskIops",
 "value": 0
 }
],
 "provider": {
 "uuid": "10545c15-7687ef26",
 "displayName": "QS2:ESXDC3DS1",
 "className": "Storage"
 }
 }
]
}

```

## Creating a Reservation

To create a reservation, you POST an input DTO that defines the reservation's:

- **action:** The action type: RESERVATION
- **demandName:** The display name of the reservation — If you do not specify names for the added workloads, this will be the root name for new VMs
- **placementParameters:**
  - **constraintIDs:** An array of UUIDs for placement policies that will affect the calculated placement
  - **count:** The number of workloads to place
  - **entityNames:** An array of names for the placed VMs — The array length should equal count
  - **geographicRedundancy:** If true place the workloads on unique hosts, otherwise Turbonomic can place multiple workloads on the same host
  - **templateID:** The UUID of the template that you will use to place this workload — Note that the template must include a reference to the deployment profile that you specify in `deploymentParameters`



- `expireDateTime`: When to cancel a reservation if Turbonomic cannot place all the workloads by that date — for a RESERVATION action, Turbonomic automatically sets the deploy time to equal this time
- `reserveDateTime`: The time to calculate the workload placement and create the reservation — This time cannot be earlier than the time that you POST the reservation to the API

The following listing shows an input DTO that creates a reservation. It will place four workloads, and it gives specific names of each one.

```
{
 "action": "RESERVATION",
 "demandName": "MyReservation",
 "expireDateTime": "2017-10-10T12:38:17+00:00",
 "parameters": [
 {
 "deploymentParameters": {
 "deploymentProfileID": "_c9CJMDDAEeePgeXuo0RRbw"
 },
 "placementParameters": {
 "geographicRedundancy": false,
 "count": 4,
 "entityNames": [
 "foo", "bar", "baz", "bonk"
],
 "templateID": "_UKsnkJkSEeCHcOXehzJExA"
 }
 }
],
 "reserveDateTime": "2017-05-04T18:22:12+00:00"
}
```

## Async Placement Calculation

When you POST a reservation, Turbonomic runs a plan to calculate the optimal placement of the workloads. Depending on the size of the reservation, this can take a significant amount of time. The API calculates the placement asynchronously. If your script relies on the reservation, you should verify that the reservation has completed before moving on.

When you execute a reservation, the response to your POST shows that the calculation is in progress:

```
{
 "uuid": "_0dPrYTDlEeePgeXuo0RRbw",
 "displayName": "MyReservation",
 "count": 4,
 "status": "IN_PROGRESS",
 "reserveDateTime": "Thu May 04 18:22:12 UTC 2017",
 "expireDateTime": "Tue Oct 10 12:38:17 UTC 2017"
}
```

To examine the placement results, GET the reservation using the UUID that the API gives in the response.

## Using Templates

To create a reservation you specify a template that contains workload requirements. VM Templates specify the resources that will be available to the VM, including:

- VCPUs
- Virtual Memory
- Storage
- Network Throughput
- IOPS
- IO Throughput

Note that templates can be created by users, and Turbonomic also discovers templates that are created by the management services in your environment. For example, a hypervisor or a cloud service provider typically manages a number of its own templates — Turbonomic discovers these. You should never edit a discovered template.

To see whether a template is discovered, GET the template you're interested in and look for the discovered property. This will be true or false.

## Turbonomic REST API Endpoints

---

The REST API resources give you full access to the Turbonomic software. This is a complete API that exposes the full set of Turbonomic capabilities. In fact, the product user interface is implemented as a Turbonomic client that uses this API.

While the REST API resources are documented in the SwaggerUI, this section provides extra details for the more expressive resources in the API.

To access the Turbonomic REST SwaggerUI, open a web browser to:

```
https://<Your_Turbonomic_IP>/swagger/external/index.html
```

### Utility Requests

In the Turbonomic API Guide, utility requests are the requests within an endpoint that do not return information about your environment, but provide benefits to other calls you may make. For example, `https://10.10.10.10/api/v3/settings/` is a utility request that returns a list of all possible settings managers. This information can be used to find settings on your instance, but the `/settings/` request itself does not provide any instance-specific information.

## Actions Endpoint

Instead of responding to thresholds, Turbonomic analyzes the operating conditions and constantly recommends actions to keep the entire environment within the desired state. If you execute these actions (or let Turbonomic execute them for you), the environment will maintain operating conditions that assure performance while also ensuring the lowest possible cost thanks to efficient utilization of your resources.

Using the actions endpoint, you can:

- Get a complete or filtered list of actions in your environment
- Accept or reject actions
- Get statistics related to a specified action
- Get notifications related to a specified action

### Actions Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

### Getting the Valid `actionMode` parameters for a Specified Entity and Action

The actions endpoint contains a utility request that shows the possible values for the `actionMode` parameter for a specified entity and action type. These can be used to set or understand the possibilities for various action settings.

To specify an entity/action type, this request takes the following parameters:

- `action_type`
  - **START:**  
Start an entity, and add it to the Turbonomic market.
  - **MOVE:**  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - **SUSPEND:**

Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.

- ADD\_PROVIDER:

This is equivalent to the START action, but for storage entities.

- CHANGE:

Move a virtual machine's storage between datastores.

- PROVISION:

Add capacity to your environment.

- RECONFIGURE:

Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.

- DELETE:

This is equivalent to the SUSPEND action, but for storage entities.

- RESERVE\_ON\_PM:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.

- RESERVE\_ON\_DS:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.

- RESIZE\_FOR\_EFFICIENCY:

A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.

- RESIZE\_FOR\_PERFORMANCE:

A resize that increases the capacity of a commodity, such as vCPU, or vMEM.

- entity\_type

- APPLICATION

- APPLICATION\_SERVER

- BUSINESS\_ACCOUNT

- BUSINESS\_UNIT

- CHASSIS

- CLOUD\_SERVICE

- CONTAINER

- CONTAINER\_POD

- DATABASE

- DATABASE\_SERVER

- DATACENTER

- DISK\_ARRAY

- DPOD

- IO\_MODULE

- LOAD\_BALANCER

- LOGICAL\_POOL

- NETWORK

- PHYSICAL\_MACHINE
- STORAGE
- STORAGE\_CONTROLLER
- SWITCH
- VIRTUAL\_APPLICATION
- VIRTUAL\_DATACENTER
- VIRTUAL\_MACHINE
- VPOD

**Example:** GET [https://10.10.10.10/api/v3/actions/availablemodes?action\\_type=PROVISION&entity\\_type=PhysicalMachine](https://10.10.10.10/api/v3/actions/availablemodes?action_type=PROVISION&entity_type=PhysicalMachine)

**Response:** An array of strings, where each string is a valid actionMode for the combination provided. In this example, a PROVISION action and the PhysicalMachine entity. An empty response body indicates that the requested entity does not have an action of that type, usually for practical reasons. For example, the Network entity cannot have a MOVE action.

```
[
 "RECOMMEND",
 "DISABLED",
 "MANUAL",
 "AUTOMATIC"
]
```

## Getting a List of Entry Points for Actions

The actions endpoint contains a utility request that shows the entry points for other action-related requests. These can be used to find the UUID and details relating to a specific action.

**Example:** GET <https://10.10.10.10/api/v3/actions>

**Response:** An array of LinkApiDTO objects, where each object contains a URL to get all actions for a specified market, entity, or group.

```
{
 "links": [
 {
 "rel": "Market actions",
 "href": "https://10.10.10.10/api/v3/markets/{uuid}/actions?ascending=true"
 },
 {
 "rel": "Entity actions",
 "href": "https://10.10.10.10/api/v3/entities/{uuid}/actions?ascending=false"
 },
 {
 "rel": "Group actions",
 "href": "https://10.10.10.10/api/v3/groups/{uuid}/actions?ascending=true"
 }
]
}
```

## Actions Requests

Action requests handle acceptance, rejection, filtering, and exploration of actions on your Turbonomic appliance.

If one or more of the entities in the scope of your request no longer exists in the topology, the ActionApiDTO for such entities will contain minimal information about the entities.

## Getting a Specified Action

Gets the action details, including the creation time and target entity information, for a specified action UUID.

**Example:** GET [https://10.10.10.10/api/v3/actions/\\_6QsKkaNWEeiYWYj--U8W5g/details](https://10.10.10.10/api/v3/actions/_6QsKkaNWEeiYWYj--U8W5g/details)

**Response:** The full ActionApiDTO for the requested action:

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/_6QsKkaNWEeiYWYj--U8W5g"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2218532354592"
 }
],
 "uuid": "_6QsKkaNWEeiYWYj--U8W5g",
 "createTime": "2018-08-19T07:40:57-04:00",
 "actionType": "MOVE",
 "actionState": "PENDING_ACCEPT",
 "actionMode": "MANUAL",
 "details": "Move VirtualMachine Jboss-EAP-10.203 from dc17-host-01.mycorp.com to dc17-host-02.mycorp.com",
 "importance": 7.4150376,
 "target": {
 "uuid": "421d8db4-3fc6-440b-0575-11383c020c6c",
 "displayName": "Jboss-EAP-10.203",
 "className": "VirtualMachine",
 "aspects": {
 "virtualMachineAspect": {
 "os": "CentOS 4/5/6/7 (64-bit)",
 "connectedNetworks": [
 {
 "uuid": "34f9f5364cd843aa1fe99900621ee3b51a7dc6e5",
 "displayName": "VM Network"
 }
]
 },
 "numVCPUs": 2,
 "ebsOptimized": false
 }
 },
 "environmentType": "ONPREM"
},
 "currentEntity": {
 "uuid": "34313836-3333-5553-4537-33364e385146",
 "displayName": "dc17-host-01.mycorp.com",
 "className": "PhysicalMachine",
 "environmentType": "ONPREM"
 },
 "newEntity": {
 "uuid": "34313836-3333-5553-4537-33394e43424e",
 "displayName": "dc17-host-02.mycorp.com",
 "className": "PhysicalMachine",
 "environmentType": "ONPREM"
 },
 "risk": {
 "uuid": "_CizZoKN2EeiYWYj--U8W5g",
 "subCategory": "Efficiency Improvement",
 "description": "dc17-host-01.mycorp.com can be suspended for efficiency",
 "severity": "MINOR",
 "importance": 0
 },
 "actionID": 2218532354592
}
```

## Getting a Filtered List of Actions

Gets a list of actions, given a scope UUID and filtered by an ActionApiInputDTO using one or more of the following criteria:

- actionModeList

- DISABLED:  
Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
- RECOMMENDED:  
Recommend the action so a user can execute it via the given hypervisor or by other external means
- MANUAL:  
Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
- AUTOMATED:  
You have directed Turbonomic to execute the action automatically.
- COLLECTION (**UNUSED**):  
Legacy action mode, no longer used.
- actionStateList
  - DISABLED:  
An action whose policy has been disabled.
  - RECOMMEND:  
An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - PENDING\_ACCEPT:  
When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - ACCEPTED:  
An action that has been accepted, but is not yet in progress.
  - IN\_PROGRESS:  
An action that is being executed.
  - SUCCEEDED:  
An action that was accepted and completed successfully.
  - FAILED:  
An action that was accepted and did not complete successfully.
  - REJECTED:  
When the action mode is set to manual, an action that has been rejected by the user.
  - CLEARED:  
When the action mode is set to manual, an action that is no longer recommended by the market.
  - ACCOUNTING:  
For cloud entities, an action to resize in order to use a different instance template.
  - QUEUED:  
When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.
- actionTypesList
  - START:  
Start an entity, and add it to the Turbonomic market.
  - MOVE:

Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.

– SUSPEND:

Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.

– ADD PROVIDER:

This is equivalent to the START action, but for storage entities.

– CHANGE:

Move a virtual machine's storage between datastores.

– PROVISION:

Add capacity to your environment.

– RECONFIGURE:

Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.

– DELETE:

This is equivalent to the SUSPEND action, but for storage entities.

– RESERVE\_ON\_PM:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.

– RESERVE\_ON\_DS:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.

– RESIZE\_FOR\_EFFICIENCY:

A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.

– RESIZE\_FOR\_PERFORMANCE:

A resize that increases the capacity of a commodity, such as vCPU, or vMEM.

• cleared

Default: `false`.

When `true`, actions that were removed from the action list before being executed will be returned.

• costType

– Saving

An action that will decrease cloud spending.

– Investment

An action that will increase cloud spending.

• cumulative

Default: `false`.

When `true`, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

• environmentType

– ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

– CLOUD:

Returns only actions related to entities that are part of your cloud environment.

- groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

- hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

The following commodities may be returned as the related commodity for an action:

- Access
- ActionPermit
- ActiveSessions
- ApplicationCommodity
- Ballooning
- Biclique
- BufferCommodity
- BurstBalance
- CPU
- CPUAllocation
- CPUProvisioned
- ClusterCommodity
- CollectionTime
- Connection
- Cooling
- Coupon
- CrossCloudMoveSvc
- CrossClusterMoveSvc
- DBCACHE\_HIT\_RATE
- DBMem
- DISK\_ARRAY\_ACCESS
- DSPMAccessCommodity
- DataCenterCommodity
- DatastoreCommodity
- DrsSegmentationCommodity



Extent  
Flow  
FlowAllocation  
HACommodity  
HOST\_LUN\_ACCESS  
Heap  
HotStorage  
IOThroughput  
ImageCPU  
ImageMem  
ImageStorage  
InstanceDiskSize  
InstanceDiskType  
LICENSE\_ACCESS  
LicenseCommodity  
Mem  
MemAllocation  
MemProvisioned  
Move  
NetThroughput  
NetworkCommodity  
NetworkInterfaceCount  
NetworkPolicy  
NumDisk  
NumberConsumers  
NumberConsumersPM  
NumberConsumersStorage  
PORT\_CHANNEL  
PoolCPU  
PoolMem  
PoolStorage  
Power  
ProcessingUnits  
Q1VCPU  
Q2VCPU  
Q3VCPU  
Q4VCPU  
Q5VCPU  
Q6VCPU  
Q7VCPU  
Q8VCPU  
Q16VCPU  
Q32VCPU  
Q64VCPU  
QNVCPU  
ResponseTime  
RightSizeDown  
RightSizeSVC  
RightSizeUp  
SLACommodity  
SameClusterMoveSvc

SegmentationCommodity  
ServiceLevelCluster  
SoftwareLicenseCommodity  
Space  
Storage  
StorageAccess  
StorageAllocation  
StorageAmount  
StorageClusterCommodity  
StorageLatency  
StorageProvisioned  
Swapping  
TemplateAccess  
TenancyAccess  
Threads  
Transaction  
TransactionLog  
Unknown  
VAppAccess  
VCPU  
VCPULimitQuota  
VCPURequest  
VCPURequestQuota  
VDCCommodity  
VMPMAccessCommodity  
VMem  
VMemLimitQuota  
VMemRequest  
VMemRequestQuota  
VStorage  
Zone

**Example:** POST <https://10.10.10.10/api/v3/actions>

**Example input:** In this example, a list of actions that meet the following criteria will be returned and grouped by the actionMode of the action:

- Is in state PENDING\_ACCEPT
- Is part of an on-premises environment
- Is an action for a virtual machine
- Is an action generated by the Turbonomic real-time market

```
{
 "actionInput": {
 "actionStateList": [
 "PENDING_ACCEPT"
],
 "environmentType": "ONPREM",
 "groupBy": [
 "actionModes"
]
 },
 "relatedType": "VirtualMachine",
 "scopes": [
 "Market"
]
}
```

**Response:** A list of ActionApiDTOs that meet the specified criteria.

## Accepting or Rejecting a Specified Action

This request accepts or rejects an action with the specified UUID, and takes the following parameters:

- action\_UUID (Required)

The UUID of the action.

- accept (Required)

When `true`, accepts the action. When `false`, rejects the action.

**Example:** `POST https://10.10.10.10/api/v3/actions/_QJcNd3JEej-LI4LMERow?accept=false`

**Response:** `true`, if the action was accepted.

## Actions Endpoint Tips

### GroupBy Field Criteria

When using the POST requests to query actions and action statistics, there is a `groupBy` parameter that you may use to group statistics. The valid values for this parameter differ by entity type. Here is the complete list of valid options per entity:

#### Note:

When sending an inputDTO using this field, only the value is needed.

**Example:** `"groupBy": "vmsByPMName"`

```
{
 "VirtualMachine":{
 "criteria":[
 {
 "filterType":"vmsByName"
 },
 {
 "filterType":"vmsByPMName"
 },
 {
 "filterType":"vmsByStorage"
 },
 {
 "filterType":"vmsByNetwork"
 },
 {
 "filterType":"vmsByApplication"
 },
 {
 "filterType":"vmsByDatabaseServer"
 },
 {
 "filterType":"vmsByDatabaseServerVersion"
 },
 {
 "filterType":"vmsByDC"
 },
 {
 "filterType":"vmsByVDC"
 },
 {
 "filterType":"vmsByDCnested"
 },
 {
 "filterType":"vmsByNumCPUs"
 },
 {
 "filterType":"vmsByMem"
 }
]
 }
}
```

```

 "filterType": "vmsByGuestName"
 },
 "filterType": "vmsByAltName"
 },
 "filterType": "vmsByClusterName"
 },
 "filterType": "vmsByDiskArrayName"
 },
 "filterType": "vmsByLogicalPoolName"
 },
 "filterType": "vmsByProfileName"
 },
 "filterType": "vmsByTag"
 },
 "filterType": "vmsByState"
 },
 "filterType": "vmsByBusinessAccountUuid"
 },
 "filterType": "vmsByResourceGroupUuid"
 }
]
},
"VirtualDataCenter": {
 "criteria": [
 {
 "filterType": "vdcByName"
 },
 {
 "filterType": "vdcByVDCName"
 },
 {
 "filterType": "vdcByTag"
 },
 {
 "filterType": "vdcByState"
 }
]
},
"PhysicalMachine": {
 "criteria": [
 {
 "filterType": "pmsByName"
 },
 {
 "filterType": "pmsByStorage"
 },
 {
 "filterType": "pmsByNetwork"
 },
 {
 "filterType": "pmsBySwitch"
 },
 {
 "filterType": "pmsByNumVms"
 },
 {
 "filterType": "pmsByDC"
 },
 {
 "filterType": "pmsByMem"
 },
 {
 "filterType": "pmsByNumCPUs"
 },
 {
 "filterType": "pmsByVendorName"
 },
 {
 "filterType": "pmsByCPUModel"
 },
 {
 "filterType": "pmsByModel"
 }
]
},

```

```

 {
 "filterType": "pmsByTimezone"
 },
 {
 "filterType": "pmsByClusterName"
 },
 {
 "filterType": "pmsByTag"
 },
 {
 "filterType": "pmsByState"
 }
]
},
"Storage": {
 "criteria": [
 {
 "filterType": "storageByName"
 },
 {
 "filterType": "storageByTag"
 },
 {
 "filterType": "storageByVMs"
 },
 {
 "filterType": "storageByDC"
 },
 {
 "filterType": "storageByPMCluster"
 },
 {
 "filterType": "storageByState"
 }
]
},
"Application": {
 "criteria": [
 {
 "filterType": "appsByName"
 },
 {
 "filterType": "appsByTag"
 }
]
},
"ApplicationComponent": {
 "criteria": [
 {
 "filterType": "appSrvsByName"
 }
]
},
"WebServer": {
 "criteria": [
 {
 "filterType": "wbSrvsByName"
 }
]
},
"BusinessApplication": {
 "criteria": [
 {
 "filterType": "busAppsByName"
 }
]
},
"Database": {
 "criteria": [
 {
 "filterType": "databaseByName"
 },
 {
 "filterType": "databaseByTag"
 },
 {
 "filterType": "databaseByBusinessAccountUuid"
 },
 {
 "filterType": "databaseByResourceGroupUuid"
 }
]
}
]

```

```

},
"DatabaseServer":{
 "criteria":[
 {
 "filterType":"databaseServerByName"
 },
 {
 "filterType":"databaseServerByTag"
 },
 {
 "filterType":"databaseServerByBusinessAccountUuid"
 },
 {
 "filterType":"databaseServerByEngine"
 },
 {
 "filterType":"databaseServerByEdition"
 },
 {
 "filterType":"databaseServerByVersion"
 }
]
},
"VirtualApplication":{
 "criteria":[
 {
 "filterType":"vappsByName"
 }
]
},
"Cluster":{
 "criteria":[
 {
 "filterType":"clustersByName"
 },
 {
 "filterType":"clustersByTag"
 }
]
},
"DataCenter":{
 "criteria":[
 {
 "filterType":"datacentersByName"
 },
 {
 "filterType":"datacentersByTag"
 }
]
},
"Group":{
 "criteria":[
 {
 "filterType":"groupsByName"
 },
 {
 "filterType":"groupsByTargetName"
 }
]
},
"StorageCluster":{
 "criteria":[
 {
 "filterType":"storageClustersByName"
 }
]
},
"DiskArray":{
 "criteria":[
 {
 "filterType":"diskarrayByName"
 }
]
},
"Zone":{
 "criteria":[
 {
 "filterType":"zonsByName"
 }
]
},
"Region":{

```

```

 "criteria":[
 {
 "filterType":"regsByName"
 }
]
 },
 "Network":{
 "criteria":[
 {
 "filterType":"netsByName"
 }
]
 },
 "LoadBalancer":{
 "criteria":[
 {
 "filterType":"lbsByName"
 }
]
 },
 "Chassis":{
 "criteria":[
 {
 "filterType":"chasByName"
 }
]
 },
 "IOModule":{
 "criteria":[
 {
 "filterType":"ioModuleByName"
 }
]
 },
 "StorageController":{
 "criteria":[
 {
 "filterType":"storagecontrollerByName"
 }
]
 },
 "DPod":{
 "criteria":[
 {
 "filterType":"dpodByName"
 }
]
 },
 "VPod":{
 "criteria":[
 {
 "filterType":"vpodByName"
 }
]
 },
 "LogicalPool":{
 "criteria":[
 {
 "filterType":"logicalPoolByName"
 }
]
 },
 "Switch":{
 "criteria":[
 {
 "filterType":"switchByName"
 }
]
 },
 "Container":{
 "criteria":[
 {
 "filterType":"containersByName"
 },
 {
 "filterType":"containersByVMName"
 }
]
 },
 "ContainerPod":{
 "criteria":[
 {

```

```

 "filterType": "containerpodsByName"
 },
 {
 "filterType": "containerpodsByVMName"
 }
]
},
"ServiceEntity": {
 "criteria": [
 {
 "filterType": "workloadByBusinessAccountUuid"
 },
 {
 "filterType": "workloadByResourceGroupUuid"
 }
]
},
"Workload": {
 "criteria": [
 {
 "filterType": "workloadByBusinessAccountUuid"
 },
 {
 "filterType": "workloadByResourceGroupUuid"
 }
]
},
"ResourceGroup": {
 "criteria": [
 {
 "filterType": "resourceGroupByName"
 },
 {
 "filterType": "resourceGroupByUuid"
 },
 {
 "filterType": "resourceGroupByBusinessAccountUuid"
 },
 {
 "filterType": "resourceGroupByTag"
 }
]
},
"BusinessAccount": {
 "criteria": [
 {
 "filterType": "businessAccountByName"
 },
 {
 "filterType": "businessAccountByUuid"
 },
 {
 "filterType": "subBusinessAccountOfUuid"
 },
 {
 "filterType": "businessAccountValidationStatus"
 },
 {
 "filterType": "businessAccountTargetName"
 },
 {
 "filterType": "businessAccountCloudProvider"
 }
]
},
"BillingFamily": {
 "criteria": [
 {
 "filterType": "billingFamilyByName"
 }
]
},
"ViewPod": {
 "criteria": [
 {
 "filterType": "viewPodByName"
 }
]
},
"DesktopPool": {
 "criteria": [
 {

```



```

 "filterType": "desktopPoolByName"
 },
 {
 "filterType": "desktopPoolByViewPod"
 }
]
},
"BusinessUser": {
 "criteria": [
 {
 "filterType": "businessUserByName"
 },
 {
 "filterType": "businessUserByDesktopPool"
 }
]
}
}
}

```

## Admin Endpoint

The admin endpoint is a collection of requests that enable you to perform various administrative functions. To use the admin endpoint, you must be logged into Turbonomic as an administrator user.

Using the admin endpoint, you can:

- Export diagnostics locally, and send them to Support
- Get health check results to see if there are issues related to the monitoring and analysis of your environment
- Get or set proxy settings
- Get or set logging levels for various components of your environment
- Get the current product version information and check for updates
- Load existing configuration files for use in your environment

## Admin Requests

### Exporting Diagnostics

If you are experiencing problems with Turbonomic, your support representative might request that you export diagnostic data. You can export the data and then send it to the support representative as requested.

**Example:** POST <https://10.10.10.10/api/v3/admin/exports>

**Response:** true if successful, false if an error occurs while Turbonomic creates the export. The export file is saved in the /tmp directory on your instance with a file name that begins with bkp-.

### Getting Health Check Results

Get health check results regularly to see whether issues exist that are related to the monitoring and analysis of your environment.

**Example:** GET <https://10.10.10.10/api/v3/admin/health>

**Response:** [HealthCategoryResponseDTO](#) with the health check results.

```

[
 {
 "healthCategory": "TARGET",
 "categoryDisplayName": "Targets",
 "categoryHealthState": "CRITICAL",
 "responseItems": [
 {
 "subcategory": "VALIDATION",
 "healthState": "MAJOR",
 "numberOfItems": 2,

```

```

 "recommendations": []
 },
 {
 "subcategory": "DISCOVERY",
 "healthState": "CRITICAL",
 "numberOfItems": 6,
 "recommendations": [
 {
 "errorType": "INTERNAL_PROBE_ERROR",
 "description": "{Description}"
 }
]
 },
 {
 "subcategory": "DISCOVERY",
 "healthState": "NORMAL",
 "numberOfItems": 23,
 "recommendations": []
 }
]
},
{
 "healthCategory": "ACTION",
 "categoryDisplayName": "Actions",
 "categoryHealthState": "NORMAL",
 "responseItems": [
 {
 "subcategory": "ANALYSIS",
 "healthState": "NORMAL",
 "numberOfItems": 0,
 "recommendations": []
 }
]
},
{
 "healthCategory": "COMPONENT",
 "categoryDisplayName": "Components",
 "categoryHealthState": "NORMAL",
 "responseItems": [
 {
 "subcategory": "CONTAINER PODS",
 "healthState": "NORMAL",
 "numberOfItems": 45,
 "recommendations": [
 {
 "errorType": "INFO",
 "description": "Pods that are in a running state."
 }
]
 }
]
}
]
}
]

```

The response includes the following health check categories (healthCategory):

- TARGET – see whether issues exist with the targets you added, such as validation or discovery failures.
- ACTION – see whether issues exist with analyzing your environment, such as analysis timeouts.
- COMPONENT – see whether issues exist with Turbonomic components, such as container pods that are not running.

Ideally, each healthCategory shows "categoryHealthState": "NORMAL". If you see a different value, review the issues and recommendations in the responseItems key.

Use the following parameters to filter the results:

- category  
Get results for a specific healthCategory.
- health\_state  
Get results with a specific categoryHealthState. For example, choose CRITICAL to see issues that require immediate attention.

## Getting and Setting a Proxy

Gets the current state of the proxy and proxy details (if enabled).

**Example:** GET `https://10.10.10.10/api/v3/admin/httpproxy`

**Response:** A `HttpProxyDTO` describing details of the proxy. If no proxy is configured, this request returns `"isProxyEnabled": false`.

```
{
 "isProxyEnabled":true,
 "proxyHost":"10.10.111.1",
 "userName":"UserName",
 "password":"PW"
}
```

To set the proxy, pass an `HttpProxyDTO`, making sure to include `"isProxyEnabled": true` if you want to activate the proxy immediately.

**Example:** POST `https://10.10.10.10/api/v3/admin/httpproxy`

**Example `httpProxyDTO`:**

```
{
 "isProxyEnabled":true,
 "proxyHost":"10.10.111.1",
 "portNumber":"33128",
 "userName":"ProxyUser",
 "password":"Password"
}
```

**Response:** `true` if successful, `false` if an error occurs while Turbonomic is setting the proxy.

## Getting and Setting Logging Levels by Component

You can set the level of logging for different components of the Turbonomic platform. The default logging level for all components is `INFO`. Setting more verbose logging levels can greatly increase the disk space that is required to store the log files. You normally change these settings only while you're working with a Turbonomic support representative.

The available logging levels are (in order of least to most verbose): `INFO`, `WARN`, `DEBUG`, and `TRACE`.

**Example:** GET `https://10.10.10.10/api/v3/admin/logginglevels`

**Response:** A `LoggingApiDTO` with details about your current logging level settings:

```
{
 "componentLoggingLevel":{
 "Extension":"INFO",
 "Analysis":"INFO",
 "Discovery":"INFO",
 "API":"INFO",
 "Presentation":"INFO",
 "Abstraction":"INFO",
 "Monitoring":"INFO"
 }
}
```

To set the logging level for a specific component, pass a partial `LoggingApiDTO` containing the component and logging level.

**Example:** POST `https://10.10.10.10/api/v3/admin/logginglevels`

### Example LoggingApiDTO:

```
{
 "componentLoggingLevel": {
 "Extension": "WARN"
 }
}
```

**Response:** A successful response returns your input in the response body.

### Getting the Current Version

Gets the version of each Turbonomic package that your instance is running. Set the `check_for_updates` parameter to `true` to also check for updates that might be available.

This request takes the following parameters:

```
check_for_updates
```

Check for available updates before returning version information. Default: `false`.

**Example:** GET [https://10.10.10.10/api/v3/admin/versions?check\\_for\\_updates=true](https://10.10.10.10/api/v3/admin/versions?check_for_updates=true)

**Response:** A `ProductVersionDTO` describing details about the current version and available updates. If you are running the latest version of Turbonomic, the updates value reads "No Updates Available\n"

```
{
 "versionInfo": "Turbonomic Operations Manager 6.1.7 (Build 20180813223950000)\nvmt-
platform-6.1.7-20180813223950000.i586
 ..
 "updates": "Available Packages\nvmt-bundle.i586 6.2.0-20180817180112000 vmturbo\nvmt-
config.x86_64
 ..
 "marketVersion": 2
}
```

### Loading External Configuration Files

To help with diagnosis of some issues, a support representative might want you to load configuration files into your Turbonomic appliance. Use this request only when you are working with a support representative, who can guide you through the process. The support representative specifies what kind of configuration file you are loading, and the contents of that file, by specifying the required parameters.

**Example:** POST [https://10.10.10.10/api/v3/admin/configfiles?config\\_type=TOPOLOGY&topology=abc](https://10.10.10.10/api/v3/admin/configfiles?config_type=TOPOLOGY&topology=abc)

**Response:** `true` if successful, `false` if an error occurs while you are uploading the configuration.

## Businessunits Endpoint

Business units are container objects used for modeling accounts and subscriptions in Public Cloud targets, and for cloud budgets and price adjustments.

Using the `businessunits` endpoint of the Turbonomic API, you can:

- For a specified business unit, get a list of:
  - All actions relating to the entities in the business unit
  - Cloud services and pricing models
  - All entities belonging to the business unit
  - Price adjustments

- Any related (parent, sibling, or child) business units
- Statistics relating to entities in the business unit
- Templates available
- Get the supply chain for a specified business unit
- Create a business unit
- Edit a business unit
- Edit a price adjustment
- Delete a business unit

Business units may be one of three types, described below:

- BUDGET

A business unit that has a budget, and may contain children.

- DISCOUNT

A business unit that has a discount, and may contain children.

- DISCOVERED

A business unit that has been discovered by the Turbonomic instance. These business units should not be edited or deleted, except on the discovered target.

## Businessunits Requests

### Getting Business Units

Gets a list of all business units created on or discovered by your Turbonomic instance. This request takes the following parameters:

- type

The type of business unit to return. [ BUDGET, DISCOUNT, DISCOVERED ]

- cloud\_type

The cloud provider whose business units to return. [ AWS, AZURE, GCP ]

- has\_parent

Whether to return only business units that have parent business units. Default: False. When true, all relevant business units will be returned irrespective of parent business units.

**Example:** GET [https://10.10.10.10/api/v3/businessunits?type=DISCOVERED&cloud\\_type=AWS](https://10.10.10.10/api/v3/businessunits?type=DISCOVERED&cloud_type=AWS)

**Response:** A list of BusinessUnitApiDTOs that meet the selected criteria. In this case, discovered business units that are part of an AWS environment.

```
[
 {
 "uuid": "323871187550",
 "displayName": "Product Trust",
 "className": "BusinessAccount",
 "environmentType": "CLOUD",
 "master": false,
 "hasRelatedTarget": true,
 "budget": {
 "value": 0
 },
 "businessUnitType": "DISCOVERED",
 "cloudType": "AWS",
 "targets": [
 {
 "uuid": "_6glgQEEhEemQ645gSx8RYw",
 "displayName": "PT-AWS",
 "type": "AWS"
 }
]
 }
]
```

```

 }
],
 "costPrice": 0,
 "severity": "Critical",
 "membersCount": 80,
 "memberType": "Workload",
 "related": {
 "rateCard": null,
 "priceAdjustment": null
 },
 "resourceGroupsCount": 0
},
{
 "uuid": "001844731978",
 "displayName": "Advanced Engineering",
 "className": "BusinessAccount",
 "environmentType": "CLOUD",
 "master": false,
 "hasRelatedTarget": false,
 "budget": {
 "value": 0
 },
 "businessUnitType": "DISCOVERED",
 "cloudType": "AWS",
 "costPrice": 0,
 "severity": "Normal",
 "membersCount": 0,
 "memberType": "Workload",
 "related": {
 "rateCard": null,
 "priceAdjustment": null
 },
 "resourceGroupsCount": 0
},
{
 "uuid": "192821421245",
 "displayName": "Development",
 "className": "BusinessAccount",
 "environmentType": "CLOUD",
 "master": true,
 "hasRelatedTarget": true,
 "budget": {
 "value": 0
 },
 "businessUnitType": "DISCOVERED",
 "cloudType": "AWS",
 "childrenBusinessUnits": [
 "631949720430",
 "001844731978",
 "323871187550"
],
 "targets": [
 {
 "uuid": "_cBVjOMvWEeiFJej-UwMSGg",
 "displayName": "engineering.aws.amazon.com",
 "type": "AWS"
 }
],
 "costPrice": 0,
 "severity": "Critical",
 "membersCount": 33,
 "memberType": "Workload",
 "related": {
 "rateCard": null,
 "priceAdjustment": null
 },
 "resourceGroupsCount": 0
},
{
 "uuid": "631949720430",
 "displayName": "Turbonomic SaaS",
 "className": "BusinessAccount",
 "environmentType": "CLOUD",
 "master": false,
 "hasRelatedTarget": false,
 "budget": {
 "value": 0
 },
 "businessUnitType": "DISCOVERED",
 "cloudType": "AWS",
 "costPrice": 0,
 "severity": "Normal",

```

```

 "membersCount": 0,
 "memberType": "Workload",
 "related": {
 "rateCard": null,
 "priceAdjustment": null
 },
 "resourceGroupsCount": 0
 }
]

```

## Deleting a Business Unit

To delete a business unit, pass the UUID of the business unit in the request. Note, you cannot delete DISCOVERED business units.

**Example:** DELETE <https://10.10.10.10/api/v3/businessunits/49d50ac3f8adbfe649d53e72c42dd9428d1d3835>

**Response:** A response of 200 indicates successful deletion

## Getting Cloud Service and Pricing Model Pairs for a Specified Business Unit

Gets the cloud service name and pricing model for each cloud service.

**Example:** GET <https://10.10.10.10/api/v3/businessunits/192821421245/cloudservices>

**Response:** A list of CloudServicePricingModelApiDTOs representing each cloud service available to the business unit:

```

[
 {
 "uuid": "aws::631949720430::CS::AWSCloudTrail",
 "displayName": "AWS CloudTrail",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::631949720430::CS::AmazonCloudWatch",
 "displayName": "AWS CloudWatch",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::192821421245::CS::AWSDeveloperSupport",
 "displayName": "AWS Developer Support",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::001844731978::CS::AmazonDynamoDB",
 "displayName": "AWS DynamoDB",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::631949720430::CS::AmazonEC2",
 "displayName": "AWS EC2",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::001844731978::CS::AmazonEKS",
 "displayName": "AWS EKS",
 "pricingModel": "ON_DEMAND"
 },
 ...
]

```

## Getting Price Adjustments Related to a Specified Business Unit

Gets any price adjustments affecting the specified business unit.

**Example:** GET <https://10.10.10.10/api/v3/businessunits/192821421245/priceadjustments>

**Response:** A list of BusinessUnitPriceAdjustmentApiDTOs showing the corrected prices after adjustment:

```
[
 {
 "servicePriceAdjustments": [
 {
 "uuid": "aws::192821421245::CS::AmazonCloudWatch",
 "displayName": "AWS CloudWatch",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::192821421245::CS::AWSDeveloperSupport",
 "displayName": "AWS Developer Support",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::192821421245::CS::AmazonDynamoDB",
 "displayName": "AWS DynamoDB",
 "pricingModel": "ON_DEMAND"
 },
 {
 "uuid": "aws::192821421245::CS::AmazonEC2",
 "displayName": "AWS EC2",
 "pricingModel": "ON_DEMAND",
 "templateDiscounts": [
 {
 "uuid": "aws::VMPROFILE::g3.16xlarge",
 "displayName": "g3.16xlarge",
 "family": "g3",
 "pricesPerDatacenter": [
 {
 "uuid": "aws::ap-southeast-1::DC::ap-southeast-1",
 "displayName": "aws-Asia Pacific (Singapore)",
 "price": 6.68
 },
 {
 "uuid": "aws::us-west-2::DC::us-west-2",
 "displayName": "aws-US West (Oregon)",
 "price": 4.56
 },
 {
 "uuid": "aws::ca-central-1::DC::ca-central-1",
 "displayName": "aws-Canada (Central)",
 "price": 5.664
 },
 {
 "uuid": "aws::eu-west-1::DC::eu-west-1",
 "displayName": "aws-EU (Ireland)",
 "price": 4.84
 },
 {
 "uuid": "aws::eu-west-3::DC::eu-west-3",
 "displayName": "aws-EU (Paris)",
 "price": 0
 },
 {
 "uuid": "aws::ap-northeast-1::DC::ap-northeast-1",
 "displayName": "aws-Asia Pacific (Tokyo)",
 "price": 6.32
 },
 {
 "uuid": "aws::eu-central-1::DC::eu-central-1",
 "displayName": "aws-EU (Frankfurt)",
 "price": 5.7
 },
 {
 "uuid": "aws::ap-south-1::DC::ap-south-1",
 "displayName": "aws-Asia Pacific (Mumbai)",
 "price": 0
 },
 {
 "uuid": "aws::ap-northeast-2::DC::ap-northeast-2",
 "displayName": "aws-Asia Pacific (Seoul)",
 "price": 0
 },
 {
 "uuid": "aws::us-east-1::DC::us-east-1",
 "displayName": "aws-US East (N. Virginia)",

```



```

 "price": 4.56
 },
 {
 "uuid": "aws::us-west-1::DC::us-west-1",
 "displayName": "aws-US West (N. California)",
 "price": 6.136
 },
 {
 "uuid": "aws::sa-east-1::DC::sa-east-1",
 "displayName": "aws-South America (Sao Paulo)",
 "price": 0
 },
 {
 "uuid": "aws::ap-southeast-2::DC::ap-southeast-2",
 "displayName": "aws-Asia Pacific (Sydney)",
 "price": 7.016
 },
 {
 "uuid": "aws::us-east-2::DC::us-east-2",
 "displayName": "aws-US East (Ohio)",
 "price": 4.56
 },
 {
 "uuid": "aws::eu-north-1::DC::eu-north-1",
 "displayName": "aws-EU (Stockholm)",
 "price": 0
 },
 {
 "uuid": "aws::eu-west-2::DC::eu-west-2",
 "displayName": "aws-EU (London)",
 "price": 0
 }
]
"uuid": "aws::VMPROFILE::p3.8xlarge",
"displayName": "p3.8xlarge",
"family": "p3",
"pricesPerDatacenter": [
 {
 "uuid": "aws::ap-southeast-1::DC::ap-southeast-1",
 "displayName": "aws-Asia Pacific (Singapore)",
 "price": 16.936
 },
 {
 "uuid": "aws::us-west-2::DC::us-west-2",
 "displayName": "aws-US West (Oregon)",
 "price": 12.24
 },
 {
 "uuid": "aws::ca-central-1::DC::ca-central-1",
 "displayName": "aws-Canada (Central)",
 "price": 13.464
 },
 {
 "uuid": "aws::eu-west-1::DC::eu-west-1",
 "displayName": "aws-EU (Ireland)",
 "price": 13.22
 },
 {
 "uuid": "aws::eu-west-3::DC::eu-west-3",
 "displayName": "aws-EU (Paris)",
 "price": 0
 },
 {
 "uuid": "aws::ap-northeast-1::DC::ap-northeast-1",
 "displayName": "aws-Asia Pacific (Tokyo)",
 "price": 16.776
 },
 {
 "uuid": "aws::eu-central-1::DC::eu-central-1",
 "displayName": "aws-EU (Frankfurt)",
 "price": 15.292
 },
 {
 "uuid": "aws::ap-south-1::DC::ap-south-1",
 "displayName": "aws-Asia Pacific (Mumbai)",
 "price": 0
 },
 {
 "uuid": "aws::ap-northeast-2::DC::ap-northeast-2",
 "displayName": "aws-Asia Pacific (Seoul)",

```



## Getting Templates Available to a Business Unit

Referenced Endpoint: [Templates](#)

Gets details about each template available to the specified business unit:

**Example:** GET <https://10.10.10.10/api/v3/businessunits/192821421245/templates>

**Response:** A list of TemplateApiDTOs representing the available templates for the specified business unit:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/templates/B4166E9F-589B-357D-AC74-0A534B53CE85"
 }
],
 "uuid": "B4166E9F-589B-357D-AC74-0A534B53CE85",
 "displayName": "db.x1e.32xlarge",
 "className": "DatabaseServerProfile",
 "price": 0,
 "model": "db.x1e.32xlarge",
 "dbEngine": "Oracle",
 "dbEdition": "Enterprise",
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCpu",
 "value": 128
 },
 {
 "name": "cpuSpeed",
 "units": "MHz",
 "value": 384000
 },
 {
 "name": "ioThroughputSize",
 "units": "MB/s",
 "value": 546.875
 },
 {
 "name": "memorySize",
 "units": "MB",
 "value": 3997696
 }
]
 }
],
 "storageResources": [
 {
 "stats": [
 {
 "name": "diskSize",
 "units": "GB",
 "value": 3840
 },
 {
 "name": "diskIops",
 "value": 35000
 }
]
 }
]
 },
 {
 "deploymentProfile": {
 "uuid": "5C57DCA5-72AB-34BF-A22D-06719D13182E",
 "displayName": "DEP-RDB-'ap-northeast-1'",
 "className": "ServiceCatalogItem",
 "deployParameters": [
 {
 "targetType": "AWS",
 "providers": [
 {
 "provider": {
 "discoveredBy": {
 "uuid": "_RRn00PMSEeisB7K_chUWVw",
 "displayName": "vmturbodev.aws.amazon.com",
 }
 }
 }
]
 }
]
 }
 }
]
```



```

 },
 {
 "name": "diskConsumedFactor",
 "units": "%",
 "value": 100
 }
]
},
"deploymentProfile": {
 "uuid": "aws:us-east-1:DP:DEP-fake-us-east-1",
 "displayName": "DEP-fake-us-east-1",
 "className": "ServiceCatalogItem",
 "deployParameters": [
 {
 "targetType": "AWS",
 "providers": [
 {
 "provider": {
 "discoveredBy": {
 "uuid": "_RRn00PMSEeisB7K_chUwVw",
 "displayName": "vmturbodev.aws.amazon.com",
 "type": "AWS"
 }
 }
 }
]
 }
]
},
"discovered": true,
"family": "g3"
},
...
]

```

## Entities Endpoint

Entities are the objects in your topology that Turbonomic manages. For example, VMs, hosts, datastores, and applications are all entities. Turbonomic discovers entities via targets, it monitors and analyzes their utilization of resources, and it recommends actions to change the placement, configuration, or state of entities.

Each market manages a set of entities. Using a market's UUID, you can get a list of entities managed by that market. For each entity you can get a full range of data, including the resources it buys and sells, the providers it buys resources from, actions for the entity, and other associated information.

Remember that more than one market can be resident in memory at a given time — Turbonomic maintains a real-time market, and there can also be plan markets in memory. As you access market entities, you should keep the following in mind:

- Entities in the real-time market reflect the current state in your physical environment
- You should never add or remove entities in the real-time market
- Entities in a plan market reflect a snapshot of the environment from when the market was created
- Plan markets can be scoped to a subset of your physical environment
- You *can* add or remove entities in a plan market

An interesting point about entities and markets — A plan market can contain entities that are also managed by the real-time market. These entities have the same stats in both markets. Turbonomic independently performs analysis on these entities in both markets.

Using the entities endpoint of the Turbonomic API, you can get a list of entities (from the market or group that contains the entities). For an individual entity you can get:

- Actions and related data for an entity
- Constraints assigned to the entity
- Through the entity constraints, get the available providers and consumers for the entity

- Groups the entity belongs to
- Notifications and notification data
- Workload placement policies that affect this entity
- Automation policies that affect this entity
- Stats for the entity
- The supply chain of entities that stitch directly with this entity
- Any tags that are defined for the entity

This endpoint contains references to start and end times for certain operations. For more information, including valid time formats, see [Time in the Turbonomic API](#)

## Entities Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

### Getting a List of Entry Points for Entities

The entities endpoint contains a utility request that shows the entry points you can use to get lists of entities. These can be used to find the UUID and details relating to specific entities.

**Example:** GET <https://10.10.10.10/api/v3/entities>

**Response:** An array of Link objects, where each object contains a URL to get all entities for a specified market or group.

```
{
 "links": [
 {
 "rel": "Market actions",
 "href": "https://10.10.10.10/api/v3/markets/{uuid}/entities"
 },
 {
 "rel": "Group actions",
 "href": "https://10.10.10.10/api/v3/groups/{uuid}/entities"
 }
]
}
```

### Getting a List of Entry Points for a Single Entity

If you pass an entity UUID to the entities endpoint, then you get a description of the entity, plus a list of entry points for entity data such as stats, constraints, actions, etc.

**Example:** GET <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073>

**Response:** An array of Link objects, where each object contains a URL to get information about the entity.

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073?include_aspects=false"
 },
 {
 "rel": "To Stats",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/stats"
 },
 {
 "rel": "Entity notifications",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/notifications"
 }
]
}
```

```

 },
 {
 "rel": "Entity actions",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions?
ascending=false"
 },
 {
 "rel": "To Constraints",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/
policies"
 },
 {
 "rel": "To Settings",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/
settings?include_settingspolicies=false"
 },
 {
 "rel": "To Entity List",
 "href": "https://10.10.10.10/api/v3/entities"
 }
],
 ...
}

```

## Entities Requests

### Getting Entities

*Referenced Endpoint:* [Groups](#)

To get a list of entities, start with the object that contains those entities. This is usually a market or a group. You work with individual entities via their UUIDs — A list of entities gives you a list of UUIDs that you can work with.

#### Note:

When working with groups, you should remember that a list of entities in the group is not the same as a list of group members. Assume a group of host clusters. In that case, the list of entities would be all the hosts in those clusters, while the group members would be a list of clusters, plus all the hosts in each cluster. For more information, see [Groups Endpoint](#).

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_BZbFswXUEemoItN1K8jw3Q/entities](https://10.10.10.10/api/v3/groups/_BZbFswXUEemoItN1K8jw3Q/entities)

**Response:** A list of EntityApiDTOs.

```

[
 {
 "links": [
 ...
],
 "uuid": "azure::VM::2ffc038a-7af3-4d23-86a9-211da2e4d433",
 "displayName": "e47f9c97-b44d-4461-be02-0304b97e201e",
 "className": "VirtualMachine",
 "priceIndex": 1.0067123,
 "state": "ACTIVE",
 "severity": "Critical",
 "costPrice": 0.64534247,
 "discoveredBy": {
 "uuid": "_-S8GgFYoEemHGJeet0B0nQ",
 "displayName": "758ad253-cbf5-4b18-8863-3eed0825bf07",
 "type": "Azure"
 },
 "remoteId": "azure::VM::2ffc038a-7af3-4d23-86a9-211da2e4d433",
 "providers": [
 {
 "uuid": "azure::ST::eastus2-managed_premium",
 "displayName": "azure-eastus2-Managed Premium",
 "className": "Storage"
 },
 {
 "uuid": "azure::eastus2::PM::eastus2",
 "displayName": "azure-eastus2",
 "className": "PhysicalMachine"
 }
]
 }
]

```

```

 }
],
 "consumers": [
 {
 "uuid": "azure::APP::2ffc038a-7af3-4d23-86a9-211da2e4d433",
 "displayName": "GuestLoad [e47f9c97-b44d-4461-be02-0304b97e201e]",
 "className": "Application"
 }
],
 "template": {
 "uuid": "azure::VMPROFILE::Standard_DS13_v2",
 "displayName": "Standard_DS13_v2",
 "discovered": false,
 "enableMatch": false
 },
 "environmentType": "CLOUD",
 "tags": {
 "instance_group": [
 "diego_cell"
]
 }
},
{
 ...
}
]

```

## Getting Actions Related to an Entity

*Referenced Endpoint: [Actions](#)*

If you provide the entity UUID but do not provide an action UUID, this returns a list of ActionApiDTOs for all the actions that are current for the entity. If you provide both the entity UUID and the action UUID, it returns the ActionApiDTO for the specified action.

This request takes the following parameters:

- **entity\_Uuid**  
The UUID of the entity whose actions will be returned.
- **action\_Uuid**  
The UUID of the action you want to get – Required to get a single action.
- **cursor**  
When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.
- **limit**  
The maximum number of items to return. Must be a positive integer. If not set, the server will provide a default (20).
- **order\_by**  
The field to order the results by. [ `creation_date`, `name`, `risk_category`, `savings`, `severity` ].
- **ascending**  
Default: true. When false, results will be in descending order.

Examples:

- Get a list of actions for the passed entity: GET [https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions?order\\_by=severity&ascending=true](https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions?order_by=severity&ascending=true)
- Get a single action for the entity: GET <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions/9223304219064809841>



**Response:** A list of ActionApiDTOs representing the actions that are current for the given entity, or a single ActionApiDTO for the specified action. The following listing shows the list of ActionApiDTOs wrapped in an array.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/9223304219106592017"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2560158594896"
 }
],
 "uuid": "9223304219106592017",
 "createTime": "2019-04-23T10:41:37-04:00",
 "actionType": "RIGHT_SIZE",
 "actionState": "PENDING_ACCEPT",
 "actionMode": "MANUAL",
 "details": "Scale down VMem for VirtualMachine vEOS-4 from 2 GB to 623 MB",
 "importance": 0,
 "target": {
 "uuid": "420b1eeb-6190-a9d2-6436-4327ad3ac073",
 "displayName": "vEOS-4",
 "className": "VirtualMachine",
 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 }
 },
 "aspects": {
 "virtualMachineAspect": {
 "os": "Other 2.6.x Linux (32-bit)",
 "connectedNetworks": [
 {
 "uuid": "d2da5316adfff0bf779652bbe7eb76c3307c2ab5",
 "displayName": "Infrastructure"
 },
 {
 "uuid": "bbef68e6ee0e76ba2c82960012b4ba563b78fdbb",
 "displayName": "vmnet4"
 },
 {
 "uuid": "d59c5e7e738ad9d2e2cb45aac5fe0073d36b49fb",
 "displayName": "vmnet6"
 },
 {
 "uuid": "96a42a4d438d1336e8d2c6962c976d889792af8a",
 "displayName": "vmnet8"
 },
 {
 "uuid": "067bc9d882e9681c55d964295e43134711520145",
 "displayName": "vmnet57"
 },
 {
 "uuid": "05b0ce5abc5cecc4104b26514b11618d77818c6d",
 "displayName": "vmnet59"
 }
]
 },
 "numVCPUs": 1,
 "type": "VMEntityAspectApiDTO"
 },
 "environmentType": "ONPREM"
 },
 {
 "currentEntity": {
 "uuid": "_3vIfwmXVEemoItN1K8jw3Q",
 "className": "VMem"
 },
 "newEntity": {
 "uuid": "_5fnl82XVEemoItN1K8jw3Q",
 "className": "VMem",
 "aspects": {
 "virtualMachineAspect": {
 "os": "Linux (Free)",
 "type": "VMEntityAspectApiDTO"
 }
 }
 }
 }
]
```

```

 }
 },
 "currentValue": "2097152.0",
 "newValue": "637952.0",
 "resizeToValue": "637952.0",
 "risk": {
 "uuid": "_5fr3YWXVEemoItN1K8jw3Q",
 "subCategory": "Efficiency Improvement",
 "description": "Underutilized VMem in Virtual Machine 'vEOS-4'",
 "severity": "MINOR",
 "importance": 0,
 "reasonCommodity": "VMem"
 },
 "stats": [
 {
 "name": "costPrice",
 "filters": [
 {
 "type": "savingsType",
 "value": "savings"
 }
]
 },
 {
 "units": "$",
 "value": 69.58008
 }
]
},
"currentLocation": {
 "uuid": "98e12595449d993db8861c1f94c8f86ba277f17b",
 "displayName": "Cloud",
 "className": "DataCenter"
},
"newLocation": {
 "uuid": "98e12595449d993db8861c1f94c8f86ba277f17b",
 "displayName": "Cloud",
 "className": "DataCenter"
},
"actionID": 2560158594896
},
...
]

```

## Getting a Filtered List of Actions Related to an Entity

Referenced Endpoint: [Actions](#)

Gets a list of actions, given a market UUID and filtered by an ActionApiInputDTO using one or more of the following criteria:

- actionModeList
  - DISABLED:
 

Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
  - RECOMMENDED:
 

Recommend the action so a user can execute it via the given hypervisor or by other external means
  - MANUAL:
 

Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
  - AUTOMATED:
 

You have directed Turbonomic to execute the action automatically.
  - COLLECTION (**UNUSED**):
 

Legacy action mode, no longer used.
- actionStateList
  - DISABLED:

- An action whose policy has been disabled.
- RECOMMEND:
  - An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
- PENDING\_ACCEPT:
  - When the action mode is set to manual, a recommended action that has not been accepted or cleared.
- ACCEPTED:
  - An action that has been accepted, but is not yet in progress.
- IN\_PROGRESS:
  - An action that is being executed.
- SUCCEEDED:
  - An action that was accepted and completed successfully.
- FAILED:
  - An action that was accepted and did not complete successfully.
- REJECTED:
  - When the action mode is set to manual, an action that has been rejected by the user.
- CLEARED:
  - When the action mode is set to manual, an action that is no longer recommended by the market.
- ACCOUNTING:
  - For cloud entities, an action to resize in order to use a different instance template.
- QUEUED:
  - When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.
- actionTypesList
  - START:
    - Start an entity, and add it to the Turbonomic market.
  - MOVE:
    - Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - SUSPEND:
    - Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.
  - ADD\_PROVIDER:
    - This is equivalent to the START action, but for storage entities.
  - CHANGE:
    - Move a virtual machine's storage between datastores.
  - PROVISION:
    - Add capacity to your environment.
  - RECONFIGURE:

Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.

– DELETE:

This is equivalent to the SUSPEND action, but for storage entities.

– RESERVE\_ON\_PM:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.

– RESERVE\_ON\_DS:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.

– RESIZE\_FOR EFFICIENCY:

A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.

– RESIZE\_FOR PERFORMANCE:

A resize that increases the capacity of a commodity, such as vCPU, or vMEM.

• cleared

Default: false.

When true, actions that were removed from the action list before being executed will be returned.

• costType

– Saving

An action that will decrease cloud spending.

– Investment

An action that will increase cloud spending.

• cumulative

Default: false.

When true, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

• environmentType

– ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

– CLOUD:

Returns only actions related to entities that are part of your cloud environment.

• groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

• hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

• relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

This request takes the following parameters:

- entity\_Uuid

The UUID of the market whose actions will be returned.

- cursor

When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.

- limit

The maximum number of items to return. Must be a positive integer. If not set, the server will provide a default (20).

- order\_by

The field to order the results by: [ creation\_date, name, risk\_category, savings, severity ].

- ascending

Default: true. When false, results will be in descending order.

**Example:** POST [https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions?order\\_by=severity&ascending=true](https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions?order_by=severity&ascending=true)

**Example input:** In this example, a list of actions that meet the following criteria will be returned and grouped by the actionMode of the action, in order of severity:

- Is in state PENDING\_ACCEPT
- Is part of an on-premises environment
- Is an action for a virtual machine

```
{
 "actionInput": {
 "actionStateList": [
 "PENDING_ACCEPT"
],
 "environmentType": "ONPREM",
 "groupBy": [
 "actionModes"
]
 },
 "relatedType": "VirtualMachine",
 "scopes": [
 "Market"
]
}
```

**Response:** A list of ActionApiDTOs that meet the specified criteria.

## Getting Statistics for a Filtered List of Actions Of an Entity

*Referenced Endpoint:* [Actions](#)

### *Referenced Endpoint: Stats*

Gets a list of statistics for a filtered list of actions that are current for the passed entity. Statistics returned include the number of actions, total savings/investment, and the number of entities affected by the actions. In addition to the Entity UUID, provide an ActionApiInputDTO using one or more of the following criteria:

- actionModeList
  - DISABLED:  
Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
  - RECOMMENDED:  
Recommend the action so a user can execute it via the given hypervisor or by other external means
  - MANUAL:  
Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
  - AUTOMATED:  
You have directed Turbonomic to execute the action automatically.
  - COLLECTION (**UNUSED**):  
Legacy action mode, no longer used.
- actionStateList
  - DISABLED:  
An action whose policy has been disabled.
  - RECOMMEND:  
An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - PENDING\_ACCEPT:  
When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - ACCEPTED:  
An action that has been accepted, but is not yet in progress.
  - IN\_PROGRESS:  
An action that is being executed.
  - SUCCEEDED:  
An action that was accepted and completed successfully.
  - FAILED:  
An action that was accepted and did not complete successfully.
  - REJECTED:  
When the action mode is set to manual, an action that has been rejected by the user.
  - CLEARED:  
When the action mode is set to manual, an action that is no longer recommended by the market.
  - ACCOUNTING:  
For cloud entities, an action to resize in order to use a different instance template.
  - QUEUED:

When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.

- `actionTypeList`
  - `START`:  
Start an entity, and add it to the Turbonomic market.
  - `MOVE`:  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - `SUSPEND`:  
Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.
  - `ADD_PROVIDER`:  
This is equivalent to the `START` action, but for storage entities.
  - `CHANGE`:  
Move a virtual machine's storage between datastores.
  - `PROVISION`:  
Add capacity to your environment.
  - `RECONFIGURE`:  
Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a `RECONFIGURE` action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.
  - `DELETE`:  
This is equivalent to the `SUSPEND` action, but for storage entities.
  - `RESERVE_ON_PM`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.
  - `RESERVE_ON_DS`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.
  - `RESIZE_FOR_EFFICIENCY`:  
A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.
  - `RESIZE_FOR_PERFORMANCE`:  
A resize that increases the capacity of a commodity, such as vCPU, or vMEM.
- `cleared`  
Default: `false`.  
When `true`, actions that were removed from the action list before being executed will be returned.
- `costType`
  - `Saving`  
An action that will decrease cloud spending.
  - `Investment`  
An action that will increase cloud spending.

- cumulative

Default: false.

When true, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

- environmentType

- ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

- CLOUD:

Returns only actions related to entities that are part of your cloud environment.

- groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

- hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

**Example:** POST <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/actions/stats>

**Example input:** Get a list of stats for actions that meet the following criteria, grouped by the actionMode of the action, in order of severity:

- Is in state PENDING\_ACCEPT
- Is part of an on-premises environment
- Is an action for a virtual machine
- Is an action generated by the Turbonomic real-time market

```
{
 "actionInput": {
 "actionStateList": [
 "PENDING_ACCEPT"
],
 "environmentType": "ONPREM",
 "groupBy": [
 "actionModes"
]
 },
 "relatedType": "VirtualMachine"
}
```



**Response:** A list of StatSnapshotApiDTOs that aggregate statistics about the filtered actions.

```
[
 {
 "date": "2019-04-24T03:56:34-04:00",
 "statistics": [
 {
 "name": "numActions",
 "values": {
 "max": 1,
 "min": 1,
 "avg": 1,
 "total": 1
 },
 "value": 1
 },
 {
 "name": "costPrice",
 "filters": [
 {
 "type": "property",
 "value": "savings"
 }
],
 "units": "$",
 "values": {
 "max": 69.77539,
 "min": 69.77539,
 "avg": 69.77539,
 "total": 69.77539
 },
 "value": 69.77539
 },
 {
 "name": "numEntities",
 "values": {
 "max": 1,
 "min": 1,
 "avg": 1,
 "total": 1
 },
 "value": 1
 }
]
 }
]
```

### Getting the Constraints for placement of an entity

When it calculates where to place an entity, Turbonomic respects whatever placement constraints there might be in your environment. For example, unless you have a Merge Cluster policy in effect, VM placement is constrained to hosts in the same cluster.

This call gets a ConstraintApiDTO object, which describes these constraints for the specified entity. The data in this object is equivalent to the Placement Constraints view that you can see in the user interface. To see this view, you drill down to the policy view for a single entity. In this view you can see the entity's related providers and consumers. You can also open a view for each related entity to show the constraints that limit potential placement for the entity. Also in this view, you can test out different placement options. (For information about testing different placement options, see [Inspecting placement options within constraints for an element](#).)

In the same way, the ConstraintApiDTO contains data objects to describe the current placement relationships, as well as potential changes you could make. The DTO delivers an array of objects for each consumer or provider. Within each object:

- `relation` indicates whether the entity is a provider (bought) or consumer (sold)
- `relatedEntities` describes the current consumer or provider
- `placementOptions` show other placement you could exploit for this relationship, if you were to remove the current constraints (via merge policies or other means)

**Example:** GET <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/constraints>

**Response:** This listing shows a subset of the related entities and placement options:

```
[
 {
 "relation": "bought",
 "entityType": "PhysicalMachine",
 "relatedEntities": [
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/30333436-3638-5355-4532-313159335631?include_aspects=false"
 }
],
 "uuid": "30333436-3638-5355-4532-313159335631",
 "displayName": "hp-dl571.dev.mycorp.com",
 "className": "PhysicalMachine",
 "priceIndex": 6.577457,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "host-30",
 "environmentType": "ONPREM"
 }
],
 "numPotentialEntities": 1,
 "placementOptions": [
 {
 "constraintType": "ClusterCommodity",
 "scope": {
 "uuid": "17fe23bc346750d62dce8ba4274fa99b4739975f",
 "displayName": "Cloud\\Physical"
 },
 "target": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vCenter:vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "numPotentialEntities": 3,
 "key": "Cluster::17fe23bc346750d62dce8ba4274fa99b4739975f"
 },
 {
 "constraintType": "NetworkCommodity",
 "scope": {
 "displayName": "NetworkCommodity/vEOS-4"
 },
 "numPotentialEntities": 2,
 "key": "DistributedVirtualPortgroup:vmnet4"
 },
 {
 "constraintType": "DatastoreCommodity",
 "scope": {
 "uuid": "5a822e24-584bce98-ecb2-441ea15e2534",
 "displayName": "SCIO:ESXDC11DS2"
 },
 "target": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vCenter:vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "numPotentialEntities": 2,
 "key": "Storage::5a822e24-584bce98-ecb2-441ea15e2534"
 },
 {
 "constraintType": "DataCenterCommodity",
 "scope": {
 "uuid": "98e12595449d993db8861c1f94c8f86ba277f17b",
 "displayName": "Cloud"
 },
 "target": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vCenter:vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "numPotentialEntities": 7,
 "key": "DataCenter::98e12595449d993db8861c1f94c8f86ba277f17b"
 }
]
 }
]
```

```

 }
]
},
{
 "relation": "sold",
 "entityType": "Application",
 ...
}
]

```

### Inspecting Placement Options Within Constraints for an Element

Within the given constraints that limit placement of an entity, there are usually a number of options for placement. The current placement is just one of many.

For information about getting entity constraints, see [Getting the Constraints for Placement of an Entity](#).

When you get the constraints for an entity, the `ConstraintApiDTO` includes other placement options for that entity on its providers, and options to place other consumers on the entity. The call in this section passes these placement options and returns the respective providers or consumers that satisfy the resulting constraint changes.

Making this call is equivalent to clicking **Find More Placement Options** in the user interface. To see this in the user interface, set the scope to a single entity, display the POLICIES tab, and scroll down to the Placement Constraints section. Then click **Constraints** next to the provider you want to explore, and click **Find More Placement Options**.

**Example:** POST <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/constraints/entities>

**Example input:** In this example, we pass two placement options, that we copied from a `ConstraintApiDTO` for the entity we're inspecting. Note that each placement option in the `ConstraintApiDTO` includes a `constraintType` and a `key`. To pass a placement option, these are the only fields we need. For this example we pass a `ClusterCommodity` constraint and a `DatastoreCommodity` constraint.

```

{
 "relation": "bought",
 "placementOptions": [
 {
 "constraintType": "ClusterCommodity",
 "key": "Cluster::17fe23bc346750d62dce8ba4274fa99b4739975f"
 },
 {
 "constraintType": "DatastoreCommodity",
 "key": "Storage::5a822e24-584bce98-ecb2-441ea15e2534"
 }
]
}

```

**Response:** In the data we posted (above), the cluster commodity has three potential entities, and the datastore commodity has two potential entities. The response brings back the intersection of these three potential provider entities – Two hosts.

```

[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/30343835-3638-584d-5131-323530394e39?include_aspects=false"
 }
],
 "uuid": "30343835-3638-584d-5131-323530394e39",
 "displayName": "hp-dl569.dev.mycorp.com",
 "className": "PhysicalMachine",
 "priceIndex": 1.031998,
 "state": "ACTIVE",
 "severity": "Normal",

```

```

 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "host-3323",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/30333436-3638-5355-4532-313159335631?include_aspects=false"
 }
],
 "uuid": "30333436-3638-5355-4532-313159335631",
 "displayName": "hp-dl571.dev.mycorp.com",
 "className": "PhysicalMachine",
 "priceIndex": 6.916899,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "host-30",
 "environmentType": "ONPREM"
 }
]

```

## Getting Group Membership of an Entity

*Referenced Endpoint:* [Groups](#)

The Groups endpoint tells you which entities are in a given group. The `/entities/{entity_Uuid}/groups` call tells you which groups a given entity belongs to. You can make this call in two modes:

- Include relationship paths

In this mode the results include only those groups that are in a buy or sell relationship with the entity. For example, a VM entity might have relationships with a datacenter and a host cluster. The call would return entries for the datacenter, the host cluster, and the VM itself.

- Do not include relationship paths

In this mode the call returns every group that includes the passed entity as a member entity. This can be a large number of groups, including all the discovered groups as well as any custom groups you created. The results do not show buy/sell relationships.

This request takes the following parameters:

- `entity_Uuid`

The UUID of the entity whose actions will be returned.

- `path`

Default: `false`. When `true`, results include only related groups, and they show the providers and consumers for each group when applicable.

### Examples:

- All groups (`path = false`): GET `https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/groups?path=false`
- Buy/Sell related groups, only (`path = true`): GET `https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/groups?path=true`

**Response:** A list of related groups. For the default (`path = false`), this returns an array of group objects.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_1Slr0GddEemDFcIzuV5wcQ?include_aspects=false"
 }
],
 "uuid": "_1Slr0GddEemDFcIzuV5wcQ",
 "displayName": "CUDA_Vm_Group_vEOS",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 5,
 "membersCount": 5,
 "groupType": "VirtualMachine",
 "severity": "Normal",
 "isStatic": true,
 "logicalOperator": "OR",
 "activeEntitiesCount": 3
 },
 ...
]
```

## Getting Settings Policies Affecting an Entity

*Referenced Endpoint:* [Settings Policies](#)

This call returns an array of `SettingsPolicyApiDTOs` for all the Automation Policies that are effective for the given entity. Note that one entity can be in multiple groups, and so it can have policies that set different values for the same setting. In that case, the most conservative setting takes effect. The data returned by this call does not reflect which setting takes precedence in this case. The call merely reflects what you would see in the user interface.

**Example:** GET <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/settingspolicies>

**Response:** An array of `SettingsPolicyApiDTOs` that are effective for the entity.

```
[
 {
 "uuid": "_i-QuUGXUEemoItN1K8jw3Q",
 "displayName": "CUDA_VM_AUTOMATION_POLICY",
 "entityType": "VirtualMachine",
 "scopes": [
 {
 "uuid": "_BZbFswXUEemoItN1K8jw3Q",
 "displayName": "CUDA_VM_Group",
 "groupType": "VirtualMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
 "settingsManagers": [
 {
 "uuid": "automationmanager",
 "displayName": "Action Mode Settings",
 "category": "Automation",
 "settings": [
 {
 "uuid": "startVM",
 "displayName": "Start",
 "value": "MANUAL",
 "defaultValue": "RECOMMEND",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 }
]
 }
]
 }
]
 }
]
```

```

 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
}
]
}
],
"disabled": false,
"readOnly": false,
"default": false
}
]

```

## Getting Statistics of an Entity

*Referenced Endpoint:* [Stats](#)

Gets all statistics for the specified entity. This request takes the following parameters:

- `entity_uuid`  
The entity whose statistics will be returned.
- `q`  
Used internally by Turbonomic.

**Example:** GET <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/stats>

**Response:** A list of `StatSnapshotApiDTO`s representing the current values of the statistics.

```

[
 {
 "displayName": "AdityaReplica",
 "date": "2019-04-24T20:09:46-04:00",
 "statistics": [
 {
 "name": "priceIndex",
 "relatedEntityType": "VirtualMachine",
 "values": {
 "max": 1.01,
 "min": 1.01,
 "avg": 1.01,
 "total": 1.01
 },
 "value": 1.01
 },
 {
 "displayName": "CPUProvisioned/AdityaReplica",
 "name": "CPUProvisioned",
 "capacity": {
 "max": 3120000,
 "min": 3120000,
 "avg": 3120000,
 "total": 3120000
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
]
 }
]
 }
]

```

```

 }
],
 "units": "MHz",
 "values": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "value": 5200
},
{
 "displayName": "Swapping/AdityaReplica",
 "name": "Swapping",
 "capacity": {
 "max": 400000000,
 "min": 400000000,
 "avg": 400000000,
 "total": 400000000
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "units": "bit/sec",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
},
{
 "displayName": "NumDisk/AdityaReplica",
 "name": "NumDisk",
 "capacity": {
 "max": 2147483650,
 "min": 2147483650,
 "avg": 2147483650,
 "total": 2147483650
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "values": {
 "max": 1,
 "min": 1,
 "avg": 1,
 "total": 1
 },
 "value": 1
},
...
]
}
]

```

## Getting a Filtered List of Statistics for an Entity

*Referenced Endpoint:* [Stats](#)

Gets a list of statistics filtered by a user-created StatPeriodApiInputDTO. For more information about the criteria that make up the input for this request, see the Swagger documentation for this request, located at [https://<Your\\_Turbonomic\\_IP>/swagger/external/index.html](https://<Your_Turbonomic_IP>/swagger/external/index.html).

**Example:** POST <https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/stats>

**Example Input:** This input will return the values for the VMem, VCPU, and VStorage statistics of the specified entity, ordered first by key, then relatedEntity, then virtualDisk.

```
{
 "statistics": [
 {
 "name": "VMem",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 },
 {
 "name": "VCPU",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 },
 {
 "name": "VStorage",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 }
]
}
```

**Response:** An array of StatSnapshotApiDTOs representing the filtered statistics for the entity:

```
[
 {
 "displayName": "vEOS-4",
 "date": "2019-04-25T17:57:02-04:00",
 "statistics": [
 {
 "displayName": "vEOS-4",
 "name": "VCPU",
 "capacity": {
 "max": 2663,
 "min": 2663,
 "avg": 2663,
 "total": 2663
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "virtualDisk",
 "value": ""
 },
 {
 "type": "key",
 "value": null
 },
 {
 "type": "relation",
 "value": "sold"
 }
],
 "relatedEntity": {
 "uuid": "c0849b50a2cce61f57ea0e49e2199503e82f3f54",
 "displayName": "GuestLoad[vEOS-4]",
 "className": "Application"
 },
 "units": "MHz",
 "values": {
 "max": 143,
 "min": 91.47,
 "avg": 91.47,
 "total": 91.47
 },
 "value": 91.47
 }
]
 }
]
```



```

 },
 {
 "displayName": "vEOS-4",
 "name": "VMem",
 "capacity": {
 "max": 2097152,
 "min": 2097152,
 "avg": 2097152,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "virtualDisk",
 "value": ""
 },
 {
 "type": "key",
 "value": null
 },
 {
 "type": "relation",
 "value": "sold"
 }
],
 "relatedEntity": {
 "uuid": "c0849b50a2cce61f57ea0e49e2199503e82f3f54",
 "displayName": "GuestLoad[vEOS-4]",
 "className": "Application"
 },
 "units": "KB",
 "values": {
 "max": 188740,
 "min": 130022,
 "avg": 130022,
 "total": 130022
 },
 "value": 130022
 }
]
}
]

```

## Getting the Supply Chain of an Entity

*Referenced Endpoint:* [Supply Chains](#)

Turbonomic models your environment as a market of buyers and sellers. It maps the entities it discovers to a supply chain that represents the buy/sell relationships between these entities.

In the user interface, the default view on the Home Page shows the global supply chain. You can change scope to drill down to a supply chain for the relationships to a single entity. This is the same as the result when you get a supply chain for an entity via the API. The returned data is a `SupplyChainApiDTO` that expresses the scope for the passed entity.

The supply chain shows tiers of entity types, where depth indicates where in the supply chain levels a specific tier is. When you get a supply chain for an entity, the entity is at `depth = 0`. The depths for tiers above the entity are negative. For example, if a VM is at `depth = 0`, then the application that consumes from the VM would be at `depth = -1`. Likewise, providers to the VM would have positive depths. For example, storage that provides resources to the VM would have `depth = 1`, and a disk array that hosts the storage would have `depth = 2`.

The supply chain request takes the following parameters:

- `entity_uuid`

The entity that sets the scope of the supply chain.

- `types`

An array of strings for the entity types you want to include in the supply chain. If you leave this blank, the call returns the complete supply chain for this entity. If you specify any types, the returned supply chain contains only those types. Note that this rule also applies to the entity that sets the scope

(entity\_uuid) – If you do not include that entity's type, then the supply chain will not include that entity, even though it sets the scope of the request.

- entity\_states

An array of strings. Limit the supply chain to include only entities in the given state. Entity states are:

- ACTIVE: The entity is actively consuming resources
- EVACUATED: All consumers have been moved off of the entity, and it can be suspended or terminated
- FAILOVER: The entity is reserved for failover and will not participate in the market
- IDLE: The entity is powered off and not actively consuming market resources
- LAUNCH: The entity is starting up
- MAINTENANCE: The entity is in Maintenance mode
- NOT\_MONITORED: Turbonomic does not currently monitor the entity
- QUEUED: More than ten actions are to be executed for a single target, and this entity has pending actions in the queue
- RESOURCE\_ALLOCATION: For internal use
- RESOURCE\_RELEASE: For internal use
- SUSPEND: The entity has been placed in Suspend mode
- SUSPEND\_PENDING: The entity is making preparations to suspend, or is the next entity of its type to suspend
- TERMINATE\_PENDING: The entity is making preparations to terminate, or is the next entity of its type to terminate
- UNKNOWN: Turbonomic is unable to obtain the current state of the entity

- environment\_type

Limit the supply chain to a specific environment type. If you do not specify environment type, the supply chain includes all environments, which is the same as HYBRID. The environment types are:

- HYBRID
- CLOUD
- ONPREM

- detail\_type

The level of detail you want in your supply chain. Can be one of:

- compact: Minimal detail about the supply chain items
- entity: The supply chain items include full entity descriptions
- aspects: The supply chain items also include aspects (meta data stored with the entity)

- health

Whether to include a health summary with the entities in the supply chain. false by default.

**Example:** GET [https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/supplychains?types=VirtualMachine&types=Storage&types=DiskArray&detail\\_type=compact&health=true](https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073/supplychains?types=VirtualMachine&types=Storage&types=DiskArray&detail_type=compact&health=true)

This example gets a supply chain for a VM entity, with the following settings: including the VM, storage, and disk arrays:

- Only include the VM, its storage, and the disk arrays
- Give compact details
- Show the entity health

**Response:** The SupplyChainApiDTO for the filtered request:

```
{
 "seMap": {
 "VirtualMachine": {
 "depth": 0,
 "entitiesCount": 1,
 "healthSummary": {
 "Normal": 1
 },
 "stateSummary": {
 "ACTIVE": 1
 },
 "connectedProviderTypes": [
 "PhysicalMachine",
 "Storage",
 "VirtualDataCenter"
],
 "connectedConsumerTypes": [
 "Application"
],
 "instances": {
 "420b1eeb-6190-a9d2-6436-4327ad3ac073": {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/420b1eeb-6190-a9d2-6436-4327ad3ac073?include_aspects=false"
 }
],
 "uuid": "420b1eeb-6190-a9d2-6436-4327ad3ac073",
 "displayName": "vEOS-4",
 "className": "VirtualMachine",
 "priceIndex": 1.1463195,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "vm-100",
 "environmentType": "ONPREM"
 }
 }
 },
 "Storage": {
 "depth": 1,
 "entitiesCount": 2,
 "healthSummary": {
 "Normal": 2
 },
 "stateSummary": {
 "ACTIVE": 2
 },
 "connectedProviderTypes": [
 "DiskArray"
],
 "connectedConsumerTypes": [
 "PhysicalMachine",
 "VirtualMachine"
],
 "instances": {
 "5a822e24-584bce98-ecb2-441ea15e2534": {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/5a822e24-584bce98-ecb2-441ea15e2534?include_aspects=false"
 }
],
 "uuid": "5a822e24-584bce98-ecb2-441ea15e2534",
 "displayName": "SCIO:ESXDC11DS2",
 "className": "Storage",
 "priceIndex": 1.2797012,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_FJwg0EzHEemaXZFq01sC0g",
 "displayName": "vsphere-dc11.dev.mycorp.com",
 "type": "vCenter"
 }
 }
 }
 }
 }
}
```

```

 },
 "remoteId": "datastore-2357",
 "environmentType": "ONPREM",
 "tags": {
 "Datastore_DC11": [
 "True"
]
 }
 },
 "9bd4ee88-99c64661": {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/9bd4ee88-99c64661?
include_aspects=false"
 }
],
 "uuid": "9bd4ee88-99c64661",
 "displayName": "QS1:NFSShare",
 "className": "Storage",
 "priceIndex": 20000,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_g0QEYEaREemB8r4uTzazvA",
 "displayName": "vsphere-dc5.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "datastore-1411",
 "environmentType": "ONPREM",
 "tags": {
 "DimStor": [
 "DimStorTag"
],
 "Datastore_DC11": [
 "True"
],
 "Test_Tag": [
 "MyStorageTag"
]
 }
 }
},
"DiskArray": {
 "depth": 2,
 "entitiesCount": 2,
 "healthSummary": {
 "Normal": 2
 },
 "stateSummary": {
 "ACTIVE": 2
 },
 "connectedConsumerTypes": [
 "Storage"
],
 "instances": {
 "DiskArray-9bd4ee88-99c64661": {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/DiskArray-9bd4ee88-99c64661?
include_aspects=false"
 }
],
 "uuid": "DiskArray-9bd4ee88-99c64661",
 "displayName": "DiskArray-QS1:NFSShare",
 "className": "DiskArray",
 "priceIndex": 0,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_oQ2aIEG1EemgjaEqcE-BJw",
 "displayName": "vsphere-dc13.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "datastore-982",
 "environmentType": "ONPREM"
 },
 "DiskArray-5a822e24-584bce98-ecb2-441ea15e2534": {
 "links": [
 {

```



- Edit a group
- Delete a group
- Get a list of actions for a specified groups
- Get a list of entities in a specified group
- Get parent groups of a specified group
- Get a list of members of a specified group
- Get a list of notifications for a specified group
- Get a list of settings for a specified group
- Get a list of policies for a specified group
- Get statistics for a specified group
- Get the supply chain related to a specified group
- Get the tags belonging to a specified group

## Group Types and Class Names

Turbonomic subdivides the entities on your appliance by the following group class name, to aid in both searching and understanding the purpose of a group:

- Folder

These appear in the user interface as folders, and are for visual organization. Discovered folders represent the folder structure in the target — for example, the vCenter folder structure. In addition, Turbonomic discovers vCenter Server resource pools, and groups them into folders.

- Group

Turbonomic places discovered entities into standard groups, and users can create groups of their own with static or dynamic membership.

- RefGroup

A RefGroup can be a group of groups- for example, a group of PM cluster groups that make up a hypervisor target- or a group defined by a relationship. For example, the built-in group PMsByDatacenter is a RefGroup reflecting the relationship between the datacenters and hosts.

- Cluster / StorageCluster

Groups that correspond to discovered clusters.

- DiscoveredGroup

Groups that are defined by a target service. For example, Turbonomic can discover DRS domains that were defined in vCenter Server.

- MarketGroup

A group that is based on the infrastructure cost of that entity type. These groups are based on the Infrastructure Cost settings in the Turbonomic policies.

- StaticMetaGroup / RefMetaGroup / MetaGroup

Groups used internally by Turbonomic. These should not be modified in any way.

In addition, each group has a `groupType` that identifies the type of members that group contains. Remember that when you create a group in the user interface or via the API, you cannot add more than one type of member. The different group types include:

- Cluster

A group of clusters. For example, you could create a group of all PM clusters that include the name "Development".

- ServiceEntity

An internal group type reserved for groups that Turbonomic discovers or creates. Note that the group will still contain members of only one type.

- Entity Type
  - APPLICATION
  - APPLICATION\_SERVER
  - BUSINESS\_ACCOUNT
  - BUSINESS\_UNIT
  - CHASSIS
  - CLOUD\_SERVICE
  - CONTAINER
  - CONTAINER\_POD
  - DATABASE
  - DATABASE\_SERVER
  - DATACENTER
  - DISK\_ARRAY
  - DPOD
  - IO\_MODULE
  - LOAD\_BALANCER
  - LOGICAL\_POOL
  - NETWORK
  - PHYSICAL\_MACHINE
  - STORAGE
  - STORAGE\_CONTROLLER
  - SWITCH
  - VIRTUAL\_APPLICATION
  - VIRTUAL\_DATACENTER
  - VIRTUAL\_MACHINE
  - VPOD

Understanding the different group types helps you find the group you want. If you know the name of a specific group, then you can filter the returned list for a group of that name. Or if you want to step through all the clusters Turbonomic has discovered, you can filter all the entries with a `className` of `Cluster` or `StorageCluster`.

## Group Filter Types

Turbonomic utilizes `regex` and a `filterType` to build a group. The `filterType` is the criterion to which the `regex` will match. For example, a `filterType` of `pmsByName` will apply the `regex` to the display names of physical machines.

The available options for this parameter depend on the entity making up your group:

- VirtualMachine
  - `vmsByName`
  - `vmsByPMName`
  - `vmsByStorage`
  - `vmsByNetwork`
  - `vmsByApplication`

- vmsByDatabaseServer
- vmsByDatabaseServerVersion
- vmsByDC
- vmsByVDC
- vmsByDCnested
- vmsByNumCPUs
- vmsByMem
- vmsByGuestName
- vmsByAltName
- vmsByClusterName
- vmsByDiskArrayName
- vmsByLogicalPoolName
- vmsByTag
- vmsByState
- vmsByBusinessAccountUuid
- vmsByResourceGroupUuid
- VirtualDataCenter
  - vdcsByName
  - vdcsByVDCName
  - vdcsByTag
  - vdcsByState
- PhysicalMachine
  - pmsByName
  - pmsByStorage
  - pmsByNetwork
  - pmsBySwitch
  - pmsByNumVms
  - pmsByDC
  - pmsByMem
  - pmsByNumCPUs
  - pmsByVendorName
  - pmsByCPUModel
  - pmsByModel
  - pmsByTimezone
  - pmsByClusterName
  - pmsByTag
  - pmsByState
- Storage
  - storageByName
  - storageByTag
  - storageByVMs
  - storageByDC
  - storageByPMCluster



- storageByState
- volumeByDeletable
- Application Component
  - appSrvsByName
  - appsByName
- WebServer
  - wbSrvsByName
- BusinessApplication
  - busAppsByName
- Database
  - databaseByName
  - databaseByTag
  - databaseByBusinessAccountUuid
  - databaseByResourceGroupUuid
- DatabaseServer
  - databaseServerByName
  - databaseServerByTag
  - databaseServerByBusinessAccountUuid
  - databaseServerByEngine
  - databaseServerByEdition
  - databaseServerByVersion
- VirtualApplication
  - vappsByName
- Cluster
  - clustersByName
  - clustersByTag
- DataCenter
  - datacentersByName
  - datacentersByTag
- Group
  - groupsByName
- StorageCluster
  - storageClustersByName
- DiskArray
  - diskarrayByName
- Zone
  - zonsByName
- Region
  - regsByName
- Network
  - netsByName
- LoadBalancer

- lbsByName
- Chassis
  - chasByName
- StorageController
  - storagecontrollerByName
- DPod
  - dpodByName
- VPod
  - vpodByName
- LogicalPool
  - logicalPoolByName
- Switch
  - switchByName
- Container
  - containersByName
  - containersByPodName
  - containersByVMName
  - containersByNamespace
  - containerSpecsByName
- ContainerPod
  - containerPodsByName
  - containerPodsByNamespace
  - containerPodsByVMName
- ServiceEntity

**Note:** Groups of type `ServiceEntity` are the only groups that have multiple entity types, and are used exclusively for cloud targets.

- workloadByBusinessAccountUuid
- workloadByResourceGroupUuid
- Workload
  - workloadByBusinessAccountUuid
  - workloadByResourceGroupUuid
  - workloadControllersByName
  - workloadControllersByNamespace
  - workloadControllersByKind
  - ResourceGroup
  - resourceGroupByName
  - resourceGroupByUuid
  - resourceGroupByBusinessAccountUuid
- BusinessAccount
  - businessAccountByName
  - businessAccountByUuid
  - subBusinessAccountOfUuid

- businessAccountValidationStatus
- businessAccountTargetName
- businessAccountCloudProvider
- BillingFamily
  - billingFamilyByName
- ViewPod
  - viewPodByName
- DesktopPool
  - desktopPoolByName
  - desktopPoolByViewPod
- BusinessUser
  - businessUserByDesktopPool
  - businessUserByName
  - businessUserByViewPod
- Cloud-Specific Filters
  - computeTiersByName
  - computeTiersByMem
  - databaseTiersByName
  - databaseTiersByDBMem
  - databaseServerTiersByName
  - databaseServerTiersByVMem

Note that in the case of a discovered resource group, group members can be discovered by multiple targets. In that case, the Turbonomic API returns a single target for the group.

## Groups Requests

### Pagination for Group Requests

When you request group data, the API uses pagination to return subsets of the data. You should always use pagination when available to make sure you get your data in manageable chunks. To paginate returned data, the `limit`, `orderBy`, and `cursor` parameters control how the APIs paginate their data. If you do not specify a `limit` for the following requests, the API will paginate the data to a default limit of records:

- GET `/groups`
- GET `/groups/<groupId>/members`
- GET `/groups/<groupId>/entities`

If your scripts make these requests without setting a limit, you should be sure they check for a pagination cursor in the returned data.

For general information about pagination, see [Pagination in the Turbonomic API](#).

### Getting a List of Groups

Gets a list of all groups on your Turbonomic appliance. This will include Turbonomic internal and default groups, as well as any groups discovered from your targets, in addition to your custom groups.

**Example:** GET `https://10.10.10.10/api/v3/groups`

**Response:** A list of GroupApiDTOs representing the groups on your appliance.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_S4k98SFQEd-S4dn3K4_SSA?
include_aspects=false"
 }
],
 "uuid": "_S4k98SFQEd-S4dn3K4_SSA",
 "displayName": "Datacenters",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 12,
 "membersCount": 12,
 "groupType": "DataCenter",
 "severity": "Critical",
 "isStatic": false,
 "logicalOperator": "AND",
 "activeEntitiesCount": 44
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_S4k94SFQEd-S4dn3K4_SSA?
include_aspects=false"
 }
],
 "uuid": "_S4k94SFQEd-S4dn3K4_SSA",
 "displayName": "Chassis",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 2,
 "membersCount": 1,
 "groupType": "Chassis",
 "severity": "Normal",
 "isStatic": false,
 "logicalOperator": "AND",
 "activeEntitiesCount": 2
 },
 ...
]
```

## Creating a Group

In order to create a group in Turbonomic, provide a GroupApiDTO with a `groupType` and `isStatic`.

For a dynamic group (`isStatic = false`), you also include the `criteriaList` information:

- `caseSensitive`  
If true, the regex will be evaluated as case-sensitive.
- `expType`  
For strings, the allowable values are EQ and NEQ (Equals, Does Not Equal). For integers, you may additionally use GT, LT, GTE, LTE (Greater than, less than, greater than or equal to, and less than or equal to, respectively).
- `expVal`  
The value of the regular expression.
- `filterType`  
See **Group Filter Types** for a full list of available options

For a static group:

- Provide UUIDs for each entity that is a member of the group in the `memberUuidList`
- DO NOT provide any `criteriaList` parameters

**Example:** POST <https://10.10.10.10/api/v3/groups/>

Example input, showing the creation of a dynamic group of VMs whose names start with "AJ":

```
{
 "isStatic": false,
 "displayName": "Dallas-Dynamic",
 "memberUuidList": [],
 "criteriaList": [
 {
 "expType": "EQ",
 "expVal": "AJ.*",
 "filterType": "vmsByName",
 "caseSensitive": false
 }
],
 "groupType": "VirtualMachine"
}
```

**Response:**

```
{
 "uuid": "_4L0PcE9SEemD69d9a0uLJQ",
 "displayName": "Dallas-Dynamic",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 1,
 "membersCount": 1,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "isStatic": false,
 "logicalOperator": "AND",
 "criteriaList": [
 {
 "expVal": "AJ.*",
 "expType": "EQ",
 "filterType": "vmsByName",
 "caseSensitive": false
 }
],
 "activeEntitiesCount": 1
}
```

Example input, showing the creation of a static group of Hosts:

```
{
 "isStatic": true,
 "displayName": "Dallas-Static",
 "memberUuidList": [
 "4C4C4544-0042-5A10-804A-B1C04F384331",
 "4C4C4544-004A-3610-8057-C8C04F464331",
 "Virtual_ESX_42381e13-4256-64bb-f554-d17798ddb2ba"
],
 "criteriaList": [],
 "groupType": "PhysicalMachine"
}
```

**Response:**

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_vn_u0FJ0EemD69d9a0uLJQ?include_aspects=false"
 }
],
 "uuid": "_vn_u0FJ0EemD69d9a0uLJQ",
 "displayName": "Dallas-Static",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 3,
 "membersCount": 3,
 "groupType": "PhysicalMachine",
 "severity": "Critical",
}
```

```
"isStatic": true,
"logicalOperator": "OR",
"activeEntitiesCount": 3
}
```

## Editing a Group

In order to edit a user-created group, send the complete GroupApiDTO with your changes as the input to this request.

**Example:** PUT [https://10.10.10.10/api/v3/groups/\\_Y6BL4FDQEemD69d9a0uLJQ](https://10.10.10.10/api/v3/groups/_Y6BL4FDQEemD69d9a0uLJQ)

**Example Input:** A modified GroupApiDTO. In this case, the expVal parameter has been modified.

```
{
 "isStatic": false,
 "displayName": "Dallas-Host",
 "memberUuidList": [],
 "criterialList": [
 {
 "expType": "EQ",
 "expVal": "hp*change",
 "filterType": "pmsByName",
 "caseSensitive": false
 }
],
 "groupType": "PhysicalMachine",
 "logicalOperator": "AND"
}
```

**Response:** The full GroupApiDTO reflecting your changes:

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_Y6BL4FDQEemD69d9a0uLJQ?include_aspects=false"
 }
],
 "uuid": "_Y6BL4FDQEemD69d9a0uLJQ",
 "displayName": "Dallas-Host",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 0,
 "membersCount": 0,
 "groupType": "PhysicalMachine",
 "severity": "Normal",
 "isStatic": false,
 "logicalOperator": "AND",
 "criterialList": [
 {
 "expVal": "hp*change",
 "expType": "EQ",
 "filterType": "pmsByName",
 "caseSensitive": false
 }
],
 "activeEntitiesCount": 0
}
```

## Deleting a Group

To delete a user-created group, pass the UUID of the group in the URL of the request.

**Example:** DELETE [https://10.10.10.10/api/v3/groups/\\_vn\\_u0FJ0EemD69d9a0uLJQ](https://10.10.10.10/api/v3/groups/_vn_u0FJ0EemD69d9a0uLJQ)

**Response:** A response of 200 indicates successful deletion.

## Getting Actions Related to a Group

*Referenced Endpoint:* [Actions](#)

Gets a list of actions relating to the entities in the specified group. To get a single action, include the UUID of the action in the request. This request takes the following parameters:

- group\_Uuid

The UUID of the group whose actions will be returned.

- cursor

When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.

- limit

The maximum number of items to return. Must be a positive integer. If not set, the server will provide a default (20).

- order\_by

The field to order the results by. [ creation\_date, name, risk\_category, savings, severity ].

- ascending

Default: true. When false, results will be in descending order.

### Examples:

All actions: GET

`https://10.10.10.10/api/v3/groups/421dd4ce-a4c1-231a-71e7-64fbf598bc65/actions?limit=50&order_by=creation_date&ascending=true`

Single action: GET `https://10.10.10.10/api/v3/groups/421dd4ce-a4c1-231a-71e7-64fbf598bc65/actions/_pJg98EQaEemD69d9a0uLJQ`

**Response:** A list of ActionApiDTOs related to the specified group:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/_pJg98EQaEemD69d9a0uLJQ"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2500817580000"
 }
],
 "uuid": "_pJg98EQaEemD69d9a0uLJQ",
 "createTime": "2019-03-11T12:28:03-04:00",
 "actionType": "RIGHT_SIZE",
 "actionState": "PENDING_ACCEPT",
 "actionMode": "MANUAL",
 "details": "Scale down Memory reservation for Virtual Machine 'AJ_Test' from 2 GB to 1 GB",
 "importance": 0,
 "target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine",
 "aspects": {
 "virtualMachineAspect": {
 "os": "CentOS 4/5/6/7 (64-bit)",
 "ip": [
 "10.10.10.107"
],
 "connectedNetworks": [
 {
 "uuid": "34f9f5364cd843aa1fe99900621ee3b51a7dc6e5",
 "displayName": "VM Network"
 }
]
 }
 }
 },
 "numVCPUs": 1,
 }
]
```

```

 "ebsOptimized": false
 },
 "environmentType": "ONPREM"
},
"currentEntity": {
 "uuid": "_pJlPYUQaEemD69d9a0uLJQ",
 "className": "Mem"
},
"newEntity": {
 "uuid": "_pLCA40QaEemD69d9a0uLJQ",
 "className": "Mem",
 "aspects": {
 "virtualMachineAspect": {
 "os": "Linux (Free)",
 "ebsOptimized": false
 }
 }
},
"currentValue": "2097152.0",
"newValue": "1048576.0",
"resizeToValue": "1048576.0",
"risk": {
 "uuid": "_pL_qMEQaEemD69d9a0uLJQ",
 "subCategory": "Efficiency Improvement",
 "description": "Underutilized Memory reservation",
 "severity": "MINOR",
 "importance": 0
},
"currentLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
},
"newLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
},
"actionID": 2500817580000
},
"links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/_XdP4EEQ8EemD69d9a0uLJQ"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2501049335840"
 }
],
"uuid": "_XdP4EEQ8EemD69d9a0uLJQ",
"createTime": "2019-03-11T16:29:28-04:00",
"actionType": "RIGHT_SIZE",
"actionState": "PENDING_ACCEPT",
"actionMode": "MANUAL",
"details": "Remove VMem limit for VirtualMachine AJ_Test to maximize efficiency",
"importance": 0,
"target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine",
 "aspects": {
 "virtualMachineAspect": {
 "os": "CentOS 4/5/6/7 (64-bit)",
 "ip": [
 "10.10.10.107"
],
 "connectedNetworks": [
 {
 "uuid": "34f9f5364cd843aa1fe99900621ee3b51a7dc6e5",
 "displayName": "VM Network"
 }
],
 "numVCPUs": 1,
 "ebsOptimized": false
 }
 }
},
"environmentType": "ONPREM"
},
"currentEntity": {

```



```

 "uuid": "_XdP4EkQ8EemD69d9a0uLJQ",
 "className": "VMem"
 },
 "newEntity": {
 "uuid": "_Xnq_YUQ8EemD69d9a0uLJQ",
 "className": "VMem",
 "aspects": {
 "virtualMachineAspect": {
 "os": "Linux (Free)",
 "ebsOptimized": false
 }
 }
 },
 "currentValue": "2097152.0",
 "newValue": "0.0",
 "resizeToValue": "0.0",
 "risk": {
 "uuid": "_XnuCsUQ8EemD69d9a0uLJQ",
 "subCategory": "Efficiency Improvement",
 "description": "VMem congestion in Virtual Machine 'AJ_Test'",
 "severity": "MAJOR",
 "reasonCommodity": "VMem",
 "importance": 0
 },
 "currentLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
 },
 "newLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
 },
 "actionID": 2501049335840
}
]

```

## Getting the Entities in a Group

*Referenced Endpoint:* [Entities](#)

Gets all entities contained within a group.

**Note:** This request will not return the contents of nested groups. In order to see groups of groups, use the `groups/members` request.

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_4L0PcE9SEemD69d9a0uLJQ/entities](https://10.10.10.10/api/v3/groups/_4L0PcE9SEemD69d9a0uLJQ/entities)

**Response:** A list of EntityApiDTOs representing the entities in the specified group.

```

[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/_pJg98EQaEemD69d9a0uLJQ"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2500817580000"
 }
],
 "uuid": "_pJg98EQaEemD69d9a0uLJQ",
 "createTime": "2019-03-11T12:28:03-04:00",
 "actionType": "RIGHT_SIZE",
 "actionState": "PENDING_ACCEPT",
 "actionMode": "MANUAL",
 "details": "Scale down Memory reservation for Virtual Machine 'AJ_Test' from 2 GB to 1 GB",
 "importance": 0,
 "target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine",
 "aspects": {
 "virtualMachineAspect": {
 "os": "CentOS 4/5/6/7 (64-bit)",

```

```

 "ip": [
 "10.10.10.107"
],
 "connectedNetworks": [
 {
 "uuid": "34f9f5364cd843aa1fe99900621ee3b51a7dc6e5",
 "displayName": "VM Network"
 }
],
 "numVCPUs": 1,
 "ebsOptimized": false
 }
 },
 "environmentType": "ONPREM"
 },
 "currentEntity": {
 "uuid": "_pJlPYUQaEemD69d9a0uLJQ",
 "className": "Mem"
 },
 "newEntity": {
 "uuid": "_pLCA40QaEemD69d9a0uLJQ",
 "className": "Mem",
 "aspects": {
 "virtualMachineAspect": {
 "os": "Linux (Free)",
 "ebsOptimized": false
 }
 }
 },
 "currentValue": "2097152.0",
 "newValue": "1048576.0",
 "resizeToValue": "1048576.0",
 "risk": {
 "uuid": "_pL_qMEQaEemD69d9a0uLJQ",
 "subCategory": "Efficiency Improvement",
 "description": "Underutilized Memory reservation",
 "severity": "MINOR",
 "importance": 0
 },
 "currentLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
 },
 "newLocation": {
 "uuid": "cdcd967d1e08d20936c3aa9a0879c635214c79ee",
 "displayName": "ACM",
 "className": "DataCenter"
 },
 "actionID": 2500817580000
}
]

```

## Getting the Parent Groups of a Group

For a specified group, gets any group that the specified group belongs to.

**Note:** Turbonomic contains a default parent group for user-created groups called "My Groups". This group should not be modified or deleted.

This request takes the following parameters:

- group\_Uuid

The UUID of the group whose actions will be returned.

- path

Default: `false`. When `true`, this request returns the parent group, along with any groups traversed to obtain the parent group. For groups that are not nested, this returns the group whose ID you entered and its parent group.

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_4L0PcE9SEemD69d9a0uLJQ/groups?path=true](https://10.10.10.10/api/v3/groups/_4L0PcE9SEemD69d9a0uLJQ/groups?path=true)

**Response:** A list of GroupApiDTOs representing the group and parent group.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_S4k68yFvEd-S4dn3T4_SSA?
include_aspects=false"
 }
],
 "uuid": "_S4k68yFvEd-S4dn3T4_SSA",
 "displayName": "My Groups",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 1,
 "membersCount": 3,
 "groupType": "ServiceEntity",
 "severity": "Major",
 "isStatic": false,
 "logicalOperator": "AND",
 "activeEntitiesCount": 1
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/groups/_4L0PcE9SEemD69d9a0uLJQ?
include_aspects=false"
 }
],
 "uuid": "_4L0PcE9SEemD69d9a0uLJQ",
 "displayName": "Dallas-Dynamic",
 "className": "Group",
 "environmentType": "ONPREM",
 "entitiesCount": 1,
 "membersCount": 1,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "isStatic": false,
 "logicalOperator": "AND",
 "criteriaList": [
 {
 "expVal": "AJ.*",
 "expType": "EQ",
 "filterType": "vmsByName",
 "caseSensitive": false
 }
],
 "activeEntitiesCount": 1
 }
]
```

## Getting the Members of a Group

*Referenced Endpoint:* [Entities](#)

Gets all members of a group.

**Note:** This request returns nested groups and the member entities of each. To get only the *entities* in the specified group, use the `groups/entities` request.

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_4L0PcE9SEemD69d9a0uLJQ/members](https://10.10.10.10/api/v3/groups/_4L0PcE9SEemD69d9a0uLJQ/members)

**Response:** A list of serviceEntityApiDTOs representing the members of the group:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/183b37c68211668aecd36eadcdce9110ebc58f85?
include_aspects=false"
 }
],
 "uuid": "183b37c68211668aecd36eadcdce9110ebc58f85",
 }
]
```

```

 "displayName": "Olga",
 "className": "VirtualDataCenter",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_h3Km4Bs3EemMFrqlXNi1yQ",
 "displayName": "vsphere-dc23.corp.vmturbo.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-96",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/2332ba0f9dc14c07f11adb38e12df43c4bc8ca56?
include_aspects=false"
 }
],
 "uuid": "2332ba0f9dc14c07f11adb38e12df43c4bc8ca56",
 "displayName": "olga_do_not_touch_1",
 "className": "VirtualDataCenter",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_5kMqwNLnEeiJs9KWCz0dUQ",
 "displayName": "vsphere-dc1.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-1088",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/6668eb0940bf970d24eb13279b31a52f3c44a91f?
include_aspects=false"
 }
],
 "uuid": "6668eb0940bf970d24eb13279b31a52f3c44a91f",
 "displayName": "Gilad",
 "className": "VirtualDataCenter",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_lj_tkH39EeiKy53b3GBg1A",
 "displayName": "vsphere-dc5.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-1953",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/20b9e0acf4c6f7d5b00f744d1dc954616095f1f2?
include_aspects=false"
 }
],
 "uuid": "20b9e0acf4c6f7d5b00f744d1dc954616095f1f2",
 "displayName": "ACM",
 "className": "VirtualDataCenter",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_lj_tkH39EeiKy53b3GBg1A",
 "displayName": "vsphere-dc5.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-v1946",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {

```

```

 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/56e6cfd2bc88ef5675df9dc3e06796ceb8b6e301?
include_aspects=false"
 }
],
 "uuid": "56e6cfd2bc88ef5675df9dc3e06796ceb8b6e301",
 "displayName": "olga_do_not_touch_7",
 "className": "VirtualDataCenter",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "GRNtoNu0Eeilt9UDtI29QA",
 "displayName": "vsphere-dc7.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-1066",
 "environmentType": "ONPREM"
}
]

```

## Getting Settings of a Group

*Referenced Endpoint:* [Settings](#)

Gets both global and user-modified settings for the specified group. To get a single setting, pass the setting UUID as part of the request URL. This request takes the following parameters:

- group\_Uuid

The Uuid of the group whose settings will be returned

- include\_settingspolicies

Default: false. When true, returns the settings policies affecting the specified group, in addition to the settings.

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_WvaGk0BjEeierZ9b1bxmgg/settings?include\\_settingspolicies=false](https://10.10.10.10/api/v3/groups/_WvaGk0BjEeierZ9b1bxmgg/settings?include_settingspolicies=false)

**Response:** A list of SettingsManagerApiDTOs representing the settings, or settings and settings policies for the specified group:

```

[
 {
 "uuid": "appsettingsmanager",
 "displayName": "Application Discovery",
 "category": "AppDiscovery",
 "settings": [
 {
 "uuid": "Username",
 "displayName": "Username",
 "value": "",
 "defaultValue": "",
 "valueType": "STRING",
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
 },
 {
 "uuid": "Password",
 "displayName": "Password",
 "value": "",
 "defaultValue": "",
 "valueType": "STRING",
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
 },
 ...
]
 },
 {
 "uuid": "controlmanager",
 "displayName": "Action Workflow",
 "category": "Automation",
 "settings": [
 {
 "uuid": "moveVMWorkflow",

```

```

 "displayName": "Move Workflow",
 "value": "",
 "defaultValue": "",
 "valueType": "STRING",
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
 }
]
}
...
}

```

## Editing a Specified Setting of a Group

*Referenced Endpoint:* [Settings](#)

Edits a specified setting for the specified group. This request takes the following parameters:

- group\_Uuid  
The Uuid of the group whose setting will be edited
- settingsManager\_Uuid  
The Uuid of the settings manager whose setting will be edited
- setting\_Uuid  
The Uuid of the setting to change

**Example:** PUT [https://10.10.10.10/api/v3/groups/\\_WvaGk0BjEeierZ9b1bxmgg/settings/appsettingsmanager/Retry\\_Interval](https://10.10.10.10/api/v3/groups/_WvaGk0BjEeierZ9b1bxmgg/settings/appsettingsmanager/Retry_Interval)

### Example Input:

The new value for the setting:

```

{
 "value": "61"
}

```

**Response:** A ServiceEntityApiDTO representing the specified setting, with your changed value(s):

```

{
 "uuid": "Retry_Interval",
 "displayName": "Retry Interval [mins]",
 "value": "61",
 "defaultValue": "60.0",
 "valueType": "NUMERIC",
 "min": 0,
 "max": 50000,
 "entityType": "VirtualMachine",
 "sourceGroupName": "A_Few_ACM_VMs"
}

```

## Getting Settings Policies of a Group

*Referenced Endpoint:* [Settings Policies](#)

Gets the settings policies of the specified group.

### Example:

### Response:

## Getting Statistics of a Group

*Referenced Endpoint:* [Stats](#)

Gets all statistics for the specified group:

**Note:**

When you retrieve statistics for a cluster, using the /groups endpoint will return the aggregated statistics for all applicable cluster entities. For individual stats per entity, use the /stats endpoint.

Some statistics are only valid for certain scopes. When you request given statistics for a scope that doesn't include them, the API does not return any data. For example, the numVMs statistic is valid in only the Market and Cluster scopes – the API omits numVMs data from the response when you query other scopes.

**Example:** `https://10.10.10.10/api/v3/groups/_WvaGk0BjEeierZ9b1bxmgg/stats`

**Response:** A list of StatSnapshotApiDTOs representing the statistics for the specified group:

```
[
 {
 "displayName": "A_Few_ACM_VMs",
 "date": "2019-03-31T17:11:35-04:00",
 "statistics": [
 {
 "displayName": "(multiple)",
 "name": "priceIndex",
 "values": {
 "max": 1.44,
 "min": 1.03,
 "avg": 1.19,
 "total": 4.77
 },
 "value": 1.19
 },
 {
 "displayName": "(multiple)",
 "name": "Q1VCPU",
 "capacity": {
 "max": 20000,
 "min": 20000,
 "avg": 20000,
 "total": 60000
 },
 "reserved": {
 "max": 10000,
 "min": 10000,
 "avg": 10000,
 "total": 30000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
]
 },
 ...
]
 },
 ...
]
```

## Getting a Filtered List of Statistics of a Group

*Referenced Endpoint:* [Stats](#)

Gets a filtered list of statistics for the specified group. For more information about creating a filtered list of statistics, including example POSTs and responses, see *Filtered Statistics* in the [Statistics Endpoint](#)

**Example:** POST `https://10.10.10.10/api/v3/groups/_WvaGk0BjEeierZ9b1bxmgg/stats`

**Example input:**

```
{
 "endDate": "-1d",
 "startDate": "-1w",
 "statistics": [
```

```

{
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
]
}

```

**Response:** A list of StatSnapshotApiDTOs representing the statistics for the specified group:

```

[
 {
 "displayName": "A_Few_ACM_VMs",
 "date": "2019-03-24T00:00:00-04:00",
 "statistics": [
 {
 "name": "Mem",
 "capacity": {
 "max": 134214448,
 "min": 134214448,
 "avg": 134214448,
 "total": 1073715580
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "units": "KB",
 "values": {
 "max": 2073084,
 "min": 679782.1,
 "avg": 1124840.4,
 "total": 4499361.5
 },
 "value": 1124840.4
 },
 {
 "name": "MemAllocation",
 "capacity": {
 "max": 504346624,
 "min": 504346624,
 "avg": 504346624,
 "total": 4034772990
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "units": "KB",
 "values": {
 "max": 2073084,
 "min": 679782.1,
 "avg": 1124840.4,
 "total": 4499361.5
 },
 "value": 1124840.4
 },
 {
 "name": "Q1VCPU",
 "capacity": {
 "max": 20000,
 "min": 20000,
 "avg": 20000,
 "total": 120000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "values": {

```



```

 "max": 90,
 "min": 14,
 "avg": 41.08,
 "total": 123.23
 },
 "value": 41.08
 }
]
}
]

```

## Getting the Supply Chain of a Group

*Referenced Endpoint:* [Supply Chains](#)

Using one or more of the following parameters, you can filter the supply chain that is returned by this request:

- market\_Uuid

The UUID of the market whose supply chain will be returned

- types

The entity types whose supply chain will be returned. Choose from:

- APPLICATION
- APPLICATION\_SERVER
- BUSINESS\_ACCOUNT
- BUSINESS\_UNIT
- CHASSIS
- CLOUD\_SERVICE
- CONTAINER
- CONTAINER\_POD
- DATABASE
- DATABASE\_SERVER
- DATACENTER
- DISK\_ARRAY
- DPOD
- IO\_MODULE
- LOAD\_BALANCER
- LOGICAL\_POOL
- NETWORK
- PHYSICAL\_MACHINE
- STORAGE
- STORAGE\_CONTROLLER
- SWITCH
- VIRTUAL\_APPLICATION
- VIRTUAL\_DATACENTER
- VIRTUAL\_MACHINE
- VPOD

- entity\_states

Only return entities in the specified states. Choose from:

- ACTIVE:  
Entity is actively consuming resources.
- EVACUATED:  
Entity is ready to suspend and will not participate in the market, but will not generate a SUSPEND action.  
**Note:** This state replaces the SUSPEND state, and requires additional configuration to enable. Turbonomic does not recommend enabling this state.
- FAILOVER:  
This entity is reserved for failover and will not participate in the market.
- IDLE:  
Entity is powered off and not actively consuming market resources.
- LAUNCH:  
Entity is starting.
- MAINTENANCE:  
Entity is in maintenance mode.
- NOT\_MONITORED:  
Entity is not currently monitored by Turbonomic.
- QUEUED:  
Entity is in the process of changing from one state to another.  
**Note:** This state is internal to Turbonomic and will not appear in the UI, but may appear in an API response if the call is executed at the precise moment the entity is changing state.
- RESOURCE\_ALLOCATION:  
Cloud entity is being created.
- RESOURCE\_RELEASE:  
Cloud Entity is being deleted.
- SUSPEND:  
Entity is suspended or the guest OS is in sleep, standby, or suspended state. The entity is not removed from the environment, but will not be used in market decisions. Based on market conditions, suspended entities can be started, re-adding them to the market.
- SUSPEND\_PENDING:  
Entity is making preparations to suspend, or is the next entity of its type to suspend.
- TERMINATE\_PENDING:  
Entity is in the process of being deleted.
- UNKNOWN:  
Turbonomic is unable to obtain the current state of the entity.  
This occurs when the entity is reported to the hypervisor as having a state of DISCONNECTED, ORPHANED, or INVALID.
- environment\_type
- Only return actions that affect entities in the specified environment type. [ CLOUD, HYBRID, ONPREM ].
- detail\_type  
The level of detail returned for each item in the returned supply chain. [ compact, entity, aspects ].

- health

Return a small summary of the health of the returned supply chain.

**Example:** GET [https://10.10.10.10/api/v3/groups/\\_WvaGk0BjEeierZ9b1bxmgg/supplychains?types=PhysicalMachine&entity\\_states=ACTIVE&entity\\_states=SUSPEND&entity\\_states=MAINTENANCE&environment\\_type=ONPREM&detail\\_type=entity&health=true](https://10.10.10.10/api/v3/groups/_WvaGk0BjEeierZ9b1bxmgg/supplychains?types=PhysicalMachine&entity_states=ACTIVE&entity_states=SUSPEND&entity_states=MAINTENANCE&environment_type=ONPREM&detail_type=entity&health=true)

**Response:** A SupplychainApiDTO representing the supply chain of the specified group:

```

"seMap": {
 "PhysicalMachine": {
 "depth": 1,
 "entitiesCount": 3,
 "healthSummary": {
 "Minor": 1,
 "Normal": 2
 },
 "stateSummary": {
 "ACTIVE": 3
 },
 "connectedProviderTypes": [
 "DataCenter",
 "Storage"
],
 "instances": {
 "34313836-3333-5553-4537-33364e385146": {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/34313836-3333-5553-4537-33364e385146?include_aspects=false"
 }
],
 "uuid": "34313836-3333-5553-4537-33364e385146",
 "displayName": "dc17-host-01.dev.mycorp.com",
 "className": "PhysicalMachine",
 "priceIndex": 3.2925103,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_yXCEIF6hEeiNCKdg4ZVvVQ",
 "displayName": "vsphere-dc17.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "host-621",
 "environmentType": "ONPREM"
 },
 ...
 }
 }
}

```

## Groups Endpoint Tips

### GroupBy Field Criteria

When using the POST requests to query actions and action statistics, you can use the `groupBy` parameter to organize the results. The valid values for this parameter differ by entity type:

**Note:**

When sending an inputDTO using this field, only the value is needed.

**Example:** `"groupBy": "vmsByPMName"`

```

{
 "VirtualMachine": {
 "criteria": [

```

```

 {
 "filterType": "vmsByName"
 },
 {
 "filterType": "vmsByPMName"
 },
 {
 "filterType": "vmsByStorage"
 },
 {
 "filterType": "vmsByNetwork"
 },
 {
 "filterType": "vmsByApplication"
 },
 {
 "filterType": "vmsByDatabaseServer"
 },
 {
 "filterType": "vmsByDatabaseServerVersion"
 },
 {
 "filterType": "vmsByDC"
 },
 {
 "filterType": "vmsByVDC"
 },
 {
 "filterType": "vmsByDCnested"
 },
 {
 "filterType": "vmsByNumCPUs"
 },
 {
 "filterType": "vmsByMem"
 },
 {
 "filterType": "vmsByGuestName"
 },
 {
 "filterType": "vmsByAltName"
 },
 {
 "filterType": "vmsByClusterName"
 },
 {
 "filterType": "vmsByDiskArrayName"
 },
 {
 "filterType": "vmsByLogicalPoolName"
 },
 {
 "filterType": "vmsByProfileName"
 },
 {
 "filterType": "vmsByTag"
 },
 {
 "filterType": "vmsByState"
 },
 {
 "filterType": "vmsByBusinessAccountUuid"
 },
 {
 "filterType": "vmsByResourceGroupUuid"
 }
]
},
"VirtualDataCenter": {
 "criteria": [
 {
 "filterType": "vdcByVDCName"
 },
 {
 "filterType": "vdcByTag"
 },
 {
 "filterType": "vdcByState"
 }
]
}

```

```

]
 },
 "PhysicalMachine":{
 "criteria":[
 {
 "filterType":"pmsByName"
 },
 {
 "filterType":"pmsByStorage"
 },
 {
 "filterType":"pmsByNetwork"
 },
 {
 "filterType":"pmsBySwitch"
 },
 {
 "filterType":"pmsByNumVms"
 },
 {
 "filterType":"pmsByDC"
 },
 {
 "filterType":"pmsByMem"
 },
 {
 "filterType":"pmsByNumCPUs"
 },
 {
 "filterType":"pmsByVendorName"
 },
 {
 "filterType":"pmsByCPUModel"
 },
 {
 "filterType":"pmsByModel"
 },
 {
 "filterType":"pmsByTimezone"
 },
 {
 "filterType":"pmsByClusterName"
 },
 {
 "filterType":"pmsByTag"
 },
 {
 "filterType":"pmsByState"
 }
]
 },
 "Storage":{
 "criteria":[
 {
 "filterType":"storageByName"
 },
 {
 "filterType":"storageByTag"
 },
 {
 "filterType":"storageByVMs"
 },
 {
 "filterType":"storageByDC"
 },
 {
 "filterType":"storageByPMCluster"
 },
 {
 "filterType":"storageByState"
 }
]
 },
 "Application":{
 "criteria":[
 {
 "filterType":"appsByName"
 },
 {
 "filterType":"appsByTag"
 }
]
 }
]
}

```

```

},
"ApplicationComponent":{
 "criteria":[
 {
 "filterType":"appSrvsByName"
 }
]
},
"WebServer":{
 "criteria":[
 {
 "filterType":"wbSrvsByName"
 }
]
},
"BusinessApplication":{
 "criteria":[
 {
 "filterType":"busAppsByName"
 }
]
},
"Database":{
 "criteria":[
 {
 "filterType":"databaseByName"
 },
 {
 "filterType":"databaseByTag"
 },
 {
 "filterType":"databaseByBusinessAccountUuid"
 },
 {
 "filterType":"databaseByResourceGroupUuid"
 }
]
},
"DatabaseServer":{
 "criteria":[
 {
 "filterType":"databaseServerByName"
 },
 {
 "filterType":"databaseServerByTag"
 },
 {
 "filterType":"databaseServerByBusinessAccountUuid"
 },
 {
 "filterType":"databaseServerByEngine"
 },
 {
 "filterType":"databaseServerByEdition"
 },
 {
 "filterType":"databaseServerByVersion"
 }
]
},
"VirtualApplication":{
 "criteria":[
 {
 "filterType":"vappsByName"
 }
]
},
"Cluster":{
 "criteria":[
 {
 "filterType":"clustersByName"
 },
 {
 "filterType":"clustersByTag"
 }
]
},
"DataCenter":{
 "criteria":[
 {
 "filterType":"datacentersByName"
 }
]
},

```

```

 {
 "filterType": "datacentersByTag"
 }
],
 "Group": {
 "criteria": [
 {
 "filterType": "groupsByName"
 },
 {
 "filterType": "groupsByTargetName"
 }
]
 },
 "StorageCluster": {
 "criteria": [
 {
 "filterType": "storageClustersByName"
 }
]
 },
 "DiskArray": {
 "criteria": [
 {
 "filterType": "diskarrayByName"
 }
]
 },
 "Zone": {
 "criteria": [
 {
 "filterType": "zonsByName"
 }
]
 },
 "Region": {
 "criteria": [
 {
 "filterType": "regsByName"
 }
]
 },
 "Network": {
 "criteria": [
 {
 "filterType": "netsByName"
 }
]
 },
 "LoadBalancer": {
 "criteria": [
 {
 "filterType": "lbsByName"
 }
]
 },
 "Chassis": {
 "criteria": [
 {
 "filterType": "chasByName"
 }
]
 },
 "IOModule": {
 "criteria": [
 {
 "filterType": "ioModuleByName"
 }
]
 },
 "StorageController": {
 "criteria": [
 {
 "filterType": "storagecontrollerByName"
 }
]
 },
 "DPod": {
 "criteria": [
 {
 "filterType": "dpodByName"
 }
]
 }
}

```

```

 }
]
},
"VPod":{
 "criteria":[
 {
 "filterType":"vpodByName"
 }
]
},
"LogicalPool":{
 "criteria":[
 {
 "filterType":"logicalPoolByName"
 }
]
},
"Switch":{
 "criteria":[
 {
 "filterType":"switchByName"
 }
]
},
"Container":{
 "criteria":[
 {
 "filterType":"containersByName"
 },
 {
 "filterType":"containersByVMName"
 }
]
},
"ContainerPod":{
 "criteria":[
 {
 "filterType":"containerpodsByName"
 },
 {
 "filterType":"containerpodsByVMName"
 }
]
},
"ServiceEntity":{
 "criteria":[
 {
 "filterType":"workloadByBusinessAccountUuid"
 },
 {
 "filterType":"workloadByResourceGroupUuid"
 }
]
},
"Workload":{
 "criteria":[
 {
 "filterType":"workloadByBusinessAccountUuid"
 },
 {
 "filterType":"workloadByResourceGroupUuid"
 }
]
},
"ResourceGroup":{
 "criteria":[
 {
 "filterType":"resourceGroupByName"
 },
 {
 "filterType":"resourceGroupByUuid"
 },
 {
 "filterType":"resourceGroupByBusinessAccountUuid"
 },
 {
 "filterType":"resourceGroupByTag"
 }
]
},
"BusinessAccount":{
 "criteria":[

```



```

 {
 "filterType": "businessAccountByName"
 },
 {
 "filterType": "businessAccountByUuid"
 },
 {
 "filterType": "subBusinessAccountOfUuid"
 },
 {
 "filterType": "businessAccountValidationStatus"
 },
 {
 "filterType": "businessAccountTargetName"
 },
 {
 "filterType": "businessAccountCloudProvider"
 }
]
},
"BillingFamily": {
 "criteria": [
 {
 "filterType": "billingFamilyByName"
 }
]
},
"ViewPod": {
 "criteria": [
 {
 "filterType": "viewPodByName"
 }
]
},
"DesktopPool": {
 "criteria": [
 {
 "filterType": "desktopPoolByName"
 },
 {
 "filterType": "desktopPoolByViewPod"
 }
]
},
"BusinessUser": {
 "criteria": [
 {
 "filterType": "businessUserByName"
 },
 {
 "filterType": "businessUserByDesktopPool"
 }
]
}
}
}

```

## Licenses Endpoint

A product license determines the product features you can use, and the number of workloads that you can manage with Turbonomic. To increase the number of workloads you can manage, you can add additional licenses to your Turbonomic installation. Note that if you use multiple licenses, they must all support the same feature set.

In all circumstances, you should contact your sales representative to make sure that you get the correct license, and that you know how to install it properly.

Using the licenses endpoint, you can:

- Get a license summary
- Get a list of licenses currently added to your Turbonomic appliance
- Add a license
- Delete a license

## Licenses Requests

### Getting Licenses

Gets a list of licenses. For instances with multiple licenses, the `licenses/summary` request will return a summed total of licensed workload. For example, if you have two licenses for 10,000 workloads each, the `/licenses` request will return both `LicenseApiDTOs` in a list. The `/licenses/summary` call will return a single summed `LicenseApiDTO`, which will state that you have capacity for 20,000 workloads.

#### Examples:

- **List of Licenses:** GET `https://10.10.10.10/api/v3/licenses`
- **License Summary:** GET `https://10.10.10.10/api/v3/licenses/summary`

### Adding a License

Adds a license to your Turbonomic installation. You may optionally specify the `dryRun` parameter which, when set to true, will validate the license file but not apply it to the appliance.

#### Note:

This request requires a multipart file as input. Due to this requirement, this request cannot be executed directly from the Swagger.

**Example:** POST `https://10.10.10.10/api/v3/licenses?file=%5Bobject%20file%5D&dryRun=true`

### Deleting a License

Removes the specified license from your Turbonomic installation. Deleting a license may reduce your licensed workload capacity below your current usage. In this case, you will be unable to add more targets until more workload capacity is available.

**Example:** DELETE `https://10.10.10.10/api/v3/licenses/_I_WVcJtWEeiGEs_doVID0w`

## Logs Requests

The logs endpoint is not implemented at this time.

## Markets Endpoint

Turbonomic uses market-based analysis to perform workload management. To do this, it constructs a model of your environment, representing each entity as a buyer and seller in a market. You can access this model via a named market resource. There are two types of markets:

#### • Real-Time Market

The real-time market performs analysis and workload management on your environment. You can use the real-time market to access entities and get current or historical data about them.

The `displayName` of the real-time market is `Market`. This market should always be in the `RUNNING` state. You should not attempt to use the API to stop this market.

Turbonomic performs discovery to populate the topology it manages (the collection of entities in the real-time market). For this reason, you should never use the API add or delete entities in the real-time market.

#### • Plan Market

A plan market begins as a snapshot of the real-time market. You POST a scenario to the real-time market, and Turbonomic makes a copy of the real-time market to create a plan market. This plan market is just like the real-time market, except for any changes that were declared in the scenario. For example,

if the scenario declares a scope for the plan, then the plan market only contains entities that are within that scope.

Turbonomic runs economic cycles (buy/sell cycles) against this plan market until there are no more meaningful improvements to be made. At this point the plan run is completed.

Note that once Turbonomic creates the plan market, that market stays in memory until you delete it. Also note that you can later apply a scenario to the plan market. This is how you run a plan on a plan. The logic flow is the same, and Turbonomic creates yet another plan market to run the analysis.

This endpoint contains references to start and end times for certain operations. For more information, including valid time formats, see [Time in the Turbonomic API](#)

Using the markets endpoint of the Turbonomic API, you can:

- Get a list of markets
- Save or Stop a plan market
- Delete a plan market
- See unplaced entities in a plan market
- Get filtered and unfiltered lists of actions, entities, notifications, policies, reservations, and statistics for a particular market
- Get, edit, and delete placement policies in the real-time market

## Determining the Status of the Main Market

After restarting the Turbonomic instance, the UI and API may be available before the real-time market has finished initializing and loading the topology of your environment. To check market status, execute the GET `/markets/market_uuid` request with a market UUID of `Market`.

If information about the market is returned, it has initialized. If the real-time market is not initialized, the following response will be returned:

```
{
 "type":400,
 "exception":"com.vmturbo.api.exceptions.OperationFailedException: Infrastructure is not
 controlled yet. Could not get market.",
 "message":"Infrastructure is not controlled yet. Could not get market."
}
```

## Markets Requests

### Getting a List of Markets

Returns all the Markets currently stored on your Turbonomic appliance. To get the details of a single market, include the UUID of the market in the request.

#### Note:

One market will have the `displayName` of `Market`. This market is the real-time market. You should not add nor delete entities directly to this market using the API, nor should you attempt to execute a STOP action on this market.

#### Examples:

- All Markets: GET `https://10.10.10.10/api/v3/markets`
- Single Market: GET `https://10.10.10.10/api/v3/markets/_0x30YUg1Ed-gHc4L513y0A`

**Response:** A list of `MarketApiDTO`s representing the markets on your appliance.

```
[
 {
 "links": [
```

```

 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/markets/_0x30YUglEd-gHc4L513y0A"
 }
],
 "uuid": "_0x30YUglEd-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market",
 "state": "RUNNING",
 "unplacedEntities": false,
 "environmentType": "HYBRID"
},
...
]

```

## Stopping or Saving Plan Markets

For troubleshooting, you may need to stop or save a plan market that is taking a long time to run, or does not seem to complete. These operations should never be used on the real-time market. Note that a manually-stopped plan market may have incomplete data.

This request takes the following parameters:

- market\_Uuid  
The UUID of the market that will be stopped or saved.
- operation  
The operation to perform on the market. [ stop, save ]

**Example:** POST [https://10.10.10.10/api/v3/markets/\\_N08\\_sCi\\_EemDHtuQzfdzxw?operation=stop](https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw?operation=stop)

**Response:** The MarketApiDTO of the specified market, showing its state as STOPPING.

```

{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw"
 },
 {
 "rel": "Market notifications",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/notifications"
 },
 {
 "rel": "Market actions",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/actions?ascending=true"
 },
 {
 "rel": "To Entities",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/entities"
 },
 {
 "rel": "To Constraints",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/policies"
 },
 {
 "rel": "To Reservations",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/reservations"
 },
 {
 "rel": "To Statistics",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/stats"
 },
 {
 "rel": "To Unplaced Entities",
 "href": "https://10.10.10.10/api/v3/markets/_N08_sCi_EemDHtuQzfdzxw/unplacedentities"
 },
 {
 "rel": "To Market List",
 "href": "https://10.10.10.10/api/v3/markets"
 }
],
}

```

```

"uuid": " _N08_sCi_EemDhtuQzfdzxw",
"displayName": "ADD_WORKLOAD_administrator_1549313730114",
"className": "Market",
"state": "STOPPING",
"scenario": {
 "uuid": "2452690315520",
 "displayName": "Add Workload 1",
 "owners": [],
 "type": "ADD_WORKLOAD"
},
"unplacedEntities": false,
"runDate": "2019-02-04T15:55:30-05:00",
"environmentType": "ONPREM"
}

```

## Deleting a Plan Market

Deleting a plan market removes the market entity along with all the topological links, such as actions, notifications, entities, policies and reservations. This operation is permanent. To delete a plan market, pass the UUID of the market in the request.

This request takes the following parameters:

market\_Uuid

The UUID of the market that will be deleted.

**Example:** DELETE [https://10.10.10.10/api/v3/markets/\\_N08\\_sCi\\_EemDhtuQzfdzxw](https://10.10.10.10/api/v3/markets/_N08_sCi_EemDhtuQzfdzxw)

**Response:** The MarketApiDTO of the specified market, showing its state as DELETING.

```

{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/markets/2452692215536"
 }
],
 "uuid": "2452692215536",
 "displayName": "ADD_WORKLOAD_administrator_1549313730114",
 "className": "Market",
 "state": "DELETING",
 "scenario": {
 "uuid": "2452692204928",
 "displayName": "Add Workload 1",
 "owners": [
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/users/_4T_7kwY-Ed-WUKbEYSVIDw"
 }
],
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "displayName": "Administrator User",
 "username": "administrator",
 "roleUuid": "_4UAioQY-Ed-WUKbEYSVIDw",
 "roleName": "administrator",
 "loginProvider": "Local",
 "type": "DedicatedCustomer",
 "showSharedUserSC": false
 }
],
 "type": "ADD_WORKLOAD"
 },
 "saved": true,
 "unplacedEntities": false,
 "runDate": "2019-02-04T15:55:30-05:00",
 "runCompleteDate": "2019-02-04T15:57:26-05:00",
 "environmentType": "ONPREM"
}

```

## Getting a Filtered List of Actions for a Market

*Referenced Endpoint:* [Actions](#)

Gets a list of actions, given a market UUID and filtered by an ActionApiInputDTO using one or more of the following criteria:

- **actionModeList**
  - **DISABLED:**

Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
  - **RECOMMENDED:**

Recommend the action so a user can execute it via the given hypervisor or by other external means
  - **MANUAL:**

Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
  - **AUTOMATED:**

You have directed Turbonomic to execute the action automatically.
  - **COLLECTION (UNUSED):**

Legacy action mode, no longer used.
- **actionStateList**
  - **DISABLED:**

An action whose policy has been disabled.
  - **RECOMMEND:**

An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - **PENDING\_ACCEPT:**

When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - **ACCEPTED:**

An action that has been accepted, but is not yet in progress.
  - **IN\_PROGRESS:**

An action that is being executed.
  - **SUCCEEDED:**

An action that was accepted and completed successfully.
  - **FAILED:**

An action that was accepted and did not complete successfully.
  - **REJECTED:**

When the action mode is set to manual, an action that has been rejected by the user.
  - **CLEARED:**

When the action mode is set to manual, an action that is no longer recommended by the market.
  - **ACCOUNTING:**

For cloud entities, an action to resize in order to use a different instance template.
  - **QUEUED:**

When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.

- `actionTypeList`
  - `START`:  
Start an entity, and add it to the Turbonomic market.
  - `MOVE`:  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - `SUSPEND`:  
Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.
  - `ADD_PROVIDER`:  
This is equivalent to the `START` action, but for storage entities.
  - `CHANGE`:  
Move a virtual machine's storage between datastores.
  - `PROVISION`:  
Add capacity to your environment.
  - `RECONFIGURE`:  
Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a `RECONFIGURE` action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.
  - `DELETE`:  
This is equivalent to the `SUSPEND` action, but for storage entities.
  - `RESERVE_ON_PM`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.
  - `RESERVE_ON_DS`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.
  - `RESIZE_FOR_EFFICIENCY`:  
A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.
  - `RESIZE_FOR_PERFORMANCE`:  
A resize that increases the capacity of a commodity, such as vCPU, or vMEM.
- `cleared`  
Default: `false`.  
When `true`, actions that were removed from the action list before being executed will be returned.
- `costType`
  - `Saving`  
An action that will decrease cloud spending.
  - `Investment`  
An action that will increase cloud spending.

- cumulative

Default: false.

When true, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

- environmentType

- ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

- CLOUD:

Returns only actions related to entities that are part of your cloud environment.

- groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

- hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

This request takes the following parameters:

- market\_Uuid

The UUID of the market whose actions will be returned.

- cursor

When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.

- limit

The maximum number of items to return. Must be a positive integer. If not set, the server will provide a default (20).

- order\_by

The field to order the results by. [ creation\_date, name, risk\_category, savings, severity ].

- ascending

Default: true. When false, results will be in descending order.

**Example:** POST [https://10.10.10.10/api/v3/markets/Market/actions?order\\_by=severity&ascending=true](https://10.10.10.10/api/v3/markets/Market/actions?order_by=severity&ascending=true)



**Example input:** In this example, a list of actions that meet the following criteria will be returned and grouped by the `actionMode` of the action, in order of severity:

- Is in state `PENDING_ACCEPT`
- Is part of an on-premises environment
- Is an action for a virtual machine
- Is an action generated by the Turbonomic real-time market

```
{
 "actionInput": {
 "actionStateList": [
 "PENDING_ACCEPT"
],
 "environmentType": "ONPREM",
 "groupBy": [
 "actionModes"
]
 },
 "relatedType": "VirtualMachine",
 "scopes": [
 "Market"
]
}
```

**Response:** ActionApiDTOs meeting the specified criteria.

```
{
 ..
 "uuid": "_qFuM8Ci0EemDHtuQzfdzxw",
 "createTime": "2019-02-04T10:07:58-05:00",
 "actionType": "RIGHT_SIZE",
 "actionState": "PENDING_ACCEPT",
 "actionMode": "MANUAL",
 "details": "Scale VirtualMachine 8f0e3136-b259-43e5-41f8-0d2c64c62acc from Standard_B2ms to Standard_E2s_v3",
 "importance": 0,
 ...
}
```

## Getting Statistics for a Filtered List of Actions for a Market

*Referenced Endpoint:* [Actions](#)

*Referenced Endpoint:* [Stats](#)

Gets a list of statistics for a filtered list of actions. Statistics returned include the number of actions, total savings/investment, and the number of entities affected by the actions. In addition to the Market UUID, provide an `ActionApiInputDTO` using one or more of the following criteria:

- `actionModeList`
  - **DISABLED:**  
Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
  - **RECOMMENDED:**  
Recommend the action so a user can execute it via the given hypervisor or by other external means
  - **MANUAL:**  
Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an `/actions` API request.
  - **AUTOMATED:**  
You have directed Turbonomic to execute the action automatically.
  - **COLLECTION (UNUSED):**

Legacy action mode, no longer used.

- actionStateList
  - DISABLED:  
An action whose policy has been disabled.
  - RECOMMEND:  
An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - PENDING\_ACCEPT:  
When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - ACCEPTED:  
An action that has been accepted, but is not yet in progress.
  - IN\_PROGRESS:  
An action that is being executed.
  - SUCCEEDED:  
An action that was accepted and completed successfully.
  - FAILED:  
An action that was accepted and did not complete successfully.
  - REJECTED:  
When the action mode is set to manual, an action that has been rejected by the user.
  - CLEARED:  
When the action mode is set to manual, an action that is no longer recommended by the market.
  - ACCOUNTING:  
For cloud entities, an action to resize in order to use a different instance template.
  - QUEUED:  
When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.
- actionTypeList
  - START:  
Start an entity, and add it to the Turbonomic market.
  - MOVE:  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - SUSPEND:  
Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.
  - ADD\_PROVIDER:  
This is equivalent to the START action, but for storage entities.
  - CHANGE:  
Move a virtual machine's storage between datastores.
  - PROVISION:

- Add capacity to your environment.
  - RECONFIGURE:
    - Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.
  - DELETE:
    - This is equivalent to the SUSPEND action, but for storage entities.
  - RESERVE\_ON\_PM:
    - When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.
  - RESERVE\_ON\_DS:
    - When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.
  - RESIZE\_FOR EFFICIENCY:
    - A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.
  - RESIZE\_FOR PERFORMANCE:
    - A resize that increases the capacity of a commodity, such as vCPU, or vMEM.
- cleared
  - Default: `false`.
  - When `true`, actions that were removed from the action list before being executed will be returned.
- costType
  - Saving
    - An action that will decrease cloud spending.
  - Investment
    - An action that will increase cloud spending.
- cumulative
  - Default: `false`.
  - When `true`, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.
- environmentType
  - ONPREM:
    - Returns only actions related to entities that are part of your on-premises environment.
  - CLOUD:
    - Returns only actions related to entities that are part of your cloud environment.
- groupBy
  - You may group the actions returned by the following criteria: `[actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]`
- hasReservedInstance
  - Default: `false`.
  - When `true`, only actions that are related to reserved instances will be returned.
- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList
- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

**Example:** POST <https://10.10.10.10/api/v3/markets/Market/actions/stats>

**Example input:** In this example, a list of actions that meet the following criteria will be returned and grouped by the actionMode of the action, in order of severity:

- Is in state PENDING\_ACCEPT
- Is part of an on-premises environment
- Is an action for a virtual machine
- Is an action generated by the Turbonomic real-time market

```
{
 "actionInput":{
 "actionStateList":[
 "PENDING_ACCEPT"
],
 "environmentType":"ONPREM",
 "groupBy":[
 "actionModes"
]
 },
 "relatedType":"VirtualMachine",
 "scopes":[
 "Market"
]
}
```

**Response:** A list of StatSnapshotApiDTO containing statistics about the filtered actions.

```
[
 {
 "date":"2019-02-05T14:40:08-05:00",
 "statistics":[
 {
 "name":"numActions",
 "values":{
 "max":268,
 "min":268,
 "avg":268,
 "total":268
 },
 "value":268
 },
 {
 "name":"costPrice",
 "filters":[
 {
 "type":"property",
 "value":"savings"
 }
],
 "units":"$",
 "values":{
 "max":24100,
 "min":24100,
 "avg":24100,

```

```

 "total":24100
 },
 "value":24100
 },
 ...
]
}
]

```

## Getting All Entities Within a Market

*Referenced Endpoint:* [Entities](#)

Gets a list of all entities in the specified market. This request may be used to retrieve a list of entities used in a plan market.

**Example:** GET <https://10.10.10.10/api/v3/markets/2452692215536/entities>

**Response:** A list of `ServiceEntityApiDTOs`, representing the entities in the market. In this case, note that there is an attribute entitled `realtimeMarketReference` because these are entities from a plan market.

```

[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/_QQVtEyi_EemDHtuQzfdzxw?include_aspects=false"
 }
],
 "uuid": "_QQVtEyi_EemDHtuQzfdzxw",
 "displayName": "osp-master-2",
 "className": "VirtualMachine",
 "priceIndex": 3.6063216,
 "state": "ACTIVE",
 "severity": "Normal",
 "realtimeMarketReference": {
 "uuid": "42000c5d-9ae5-44f0-252c-daabbb423a86"
 },
 "environmentType": "ONPREM"
 },
 ...
]

```

## Getting Statistics of Entities Within a Market

*Referenced Endpoint:* [Stats](#)

*Referenced Endpoint:* [Entities](#)

Gets statistics related to entities in a specified market. For example, this request may be used to see the final calculated statistics for entities in a plan market. For more information about creating a filtered list of statistics, including example POSTs and responses, see *Filtered Statistics* in the [Statistics Endpoint](#).

## Getting Statistics of a Specified Group of Entities Within a Market

*Referenced Endpoint:* [Stats](#)

*Referenced Endpoint:* [Groups](#)

*Referenced Endpoint:* [Entities](#)

Gets statistics related to a specific group of entities in a specified market. For example, this request may be used to see the final calculated statistics for a group of hosts in a plan market. For more information about creating a filtered list of statistics, including example POSTs and responses, see *Filtered Statistics* in the [Statistics Endpoint](#).

## Getting Policies for a Market

*Referenced Endpoint:* [Policies](#)

Returns a list of all current policies affecting a specified market. For a plan market, this is a list of policies that you created, enabled, disabled, or changed in the plan configuration. For the real-time market, this is a list of policies currently available on your appliance. To get a single policy, include the UUID of the policy in the request.

### Examples:

- All policies: GET <https://10.10.10.10/api/v3/markets/2481646239616/policies>
- One policy:

```
GET https://10.10.10.10/api/v3/markets/2481646239616/policies/_z3LKSTk0EemeycL7hWgJNw
```

**Response:** A list of PolicyApiDTOs representing the policies requested.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/policies/_z3LxVzk0EemeycL7hWgJNw"
 }
],
 "uuid": "_z3LxVzk0EemeycL7hWgJNw",
 "displayName": "DalTestPolMerge",
 "type": "MERGE",
 "name": "DalTestPolMerge",
 "enabled": true,
 "commodityType": "ClusterCommodity",
 "mergeType": "Cluster",
 "mergeGroups": [
 {
 "uuid": "91ff08749fd278925d73b61b0f0f69d79c99fc3a",
 "displayName": "PMs_Beta DC6\\Cluster-1",
 "className": "Group",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 {
 "uuid": "8c17d8e79c3db85ed5d3868de622cb89187849ce",
 "displayName": "PMs_Beta DC6\\Cluster-2",
 "className": "Group",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
]
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/policies/_z3LKSTk0EemeycL7hWgJNw"
 }
],
 "uuid": "_z3LKSTk0EemeycL7hWgJNw",
 "displayName": "CommSegmentation/SegmentManager__4T_7kwY-Ed-WUKbEYSVIDw_1551123454966",
 "type": "MUST_RUN_TOGETHER",
 "name": "CommSegmentation/SegmentManager__4T_7kwY-Ed-WUKbEYSVIDw_1551123454966",
 "enabled": false,
 "capacity": 10000000000,
 "commodityType": "DrsSegmentationCommodity",
 "consumerGroup": {
 "uuid": "_z2M57Dk0EemeycL7hWgJNw",
 "displayName": "Buyers-Group-GROUP-DRS-KTG-rule/DC13-Cluster/vsphere-dc13.dev.mycorp.com",
 "className": "DiscoveredGroup",
 "groupType": "VirtualMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 "providerGroup": {
```

```

 "uuid": "_z2M57zk0EemeycL7hWgJNw",
 "displayName": "Sellers-Group-GROUP-DRS-KTG-rule/DC13-Cluster/vsphere-
dc13.dev.mycorp.com",
 "className": "DiscoveredGroup",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
},
...
]

```

## Deleting a Placement Policy for a Market

Referenced Endpoint: [Policies](#)

Deletes the specified policy, given the UUID of the policy and the market. Note that if you delete an imported policy, it will reappear after the next discovery cycle of the target it was imported from. To delete imported policies, delete them on the target directly.

**Example:** DELETE [https://10.10.10.10/api/v3/markets/2481646239616/policies/\\_z3LxVzk0EemeycL7hWgJNw](https://10.10.10.10/api/v3/markets/2481646239616/policies/_z3LxVzk0EemeycL7hWgJNw)

**Response:** A successful deletion request returns a Response Code of 200.

## Getting Reservations for a Market

Referenced Endpoint: [Reservations](#)

Returns a list of all current reservations for a specified market. For the real-time market, this is a list of reservations currently active on your appliance.

**Example:** GET <https://10.10.10.10/api/v3/markets/Market/reservations>

**Response:** A list of DemandReservationApiDTOs representing any current reservations.

```

[
 {
 "uuid": "_I6z38Tk8EemeycL7hWgJNw",
 "displayName": "DalRes2",
 "count": 1,
 "status": "RESERVED",
 "reserveDateTime": "Mon Feb 25 15:30:07 EST 2019",
 "expireDateTime": "Mon Mar 25 08:29:54 EDT 2019",
 "deployDateTime": "Tue Mar 26 08:29:54 EDT 2019",
 "reserveCount": 1,
 "deployCount": 0,
 "demandEntities": [
 {
 "uuid": "_JEZRwTk8EemeycL7hWgJNw",
 "displayName": "DalRes2_C0",
 "className": "VirtualMachine",
 "template": {
 "uuid": "T-ef47f3dbdde39f888052332cbe7d0cf2",
 "displayName": "AVG:PMs_ACM\\ACM Cluster for last 10 days",
 "className": "VirtualMachineProfile"
 },
 "placements": {
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCpu",
 "value": 2
 },
 {
 "name": "cpuSpeed",
 "units": "MHz",
 "value": 5320
 },
 {
 "name": "cpuConsumedFactor",
 "units": "%",
 "value": 4.2516437
 }
]
 }
]
 }
 }
]
 }
]

```

```

 },
 {
 "name": "memorySize",
 "units": "MB",
 "value": 6631
 },
 {
 "name": "memoryConsumedFactor",
 "units": "%",
 "value": 46.328953
 },
 {
 "name": "ioThroughput",
 "units": "MB/s",
 "value": 0.05301181
 },
 {
 "name": "networkThroughput",
 "units": "MB/s",
 "value": 0.103116125
 }
],
 "provider": {
 "uuid": "30333436-3638-5355-4532-313159335631",
 "displayName": "hp-d1571.dev.mycorp.com",
 "className": "PhysicalMachine"
 }
},
"storageResources": [
 {
 "stats": [
 {
 "name": "diskSize",
 "units": "GB",
 "value": 92.468285
 },
 {
 "name": "diskIops",
 "value": 1.6827557
 }
],
 "provider": {
 "uuid": "5a822e24-584bce98-ecb2-441ea15e2534",
 "displayName": "SCIO:ESXDC11DS2",
 "className": "Storage"
 }
 }
]
}
]

```

## Running a Scenario on a Market

*Referenced Endpoint:* [Scenarios](#)

When you run a scenario on the real-time market, it creates a plan market showing the results of the scenario. If you run a second scenario a plan market, that is called running a "plan over plan". This runs a plan new configuration on a market that is already the result of a plan run.

**Example:** POST [https://10.10.10.10/api/v3/markets/Market/scenarios/2452690315520?ignore\\_constraints=false](https://10.10.10.10/api/v3/markets/Market/scenarios/2452690315520?ignore_constraints=false)

**Response:** A MarketApiDTO showing the new state of the market- if successful, COPYING.

```

{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUD1BEemeycL7hWgJNw"
 },
 {
 "rel": "Market notifications",

```



```

 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/notifications"
 },
 {
 "rel": "Market actions",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/actions?
ascending=true"
 },
 {
 "rel": "To Entities",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/entities"
 },
 {
 "rel": "To Constraints",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/policies"
 },
 {
 "rel": "To Reservations",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/reservations"
 },
 {
 "rel": "To Statistics",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/stats"
 },
 {
 "rel": "To Unplaced Entities",
 "href": "https://10.10.10.10/api/v3/markets/_ps_iUDlBEemeycL7hWgJNw/unplacedentities"
 },
 {
 "rel": "To Market List",
 "href": "https://10.10.10.10/api/v3/markets"
 }
],
 "uuid": "_ps_iUDlBEemeycL7hWgJNw",
 "displayName": "API_Plan_Market",
 "className": "Market",
 "state": "COPYING",
 "scenario": {
 "uuid": "2452690315520",
 "displayName": "Add Workload 1",
 "owners": [
],
 "type": "ADD_WORKLOAD"
 },
 "unplacedEntities": false,
 "runDate": "2019-02-25T16:09:34-05:00",
 "environmentType": "ONPREM"
 }
}

```

## Getting Statistics for a Market

*Referenced Endpoint:* [Stats](#)

Gets statistics about the specified market. This call can be used to see statistics from the real-time market, or to see the resultant statistics in a plan market after the scenario has been applied.

**Example:** GET <https://10.10.10.10/api/v3/markets/Market/stats>

**Response:** A list of StatSnapshotApiDTOs representing the statistics of the real-time market.

```

[
 {
 "displayName": "PMs",
 "date": "2019-02-23T00:00:00-05:00",
 "statistics": [
 {
 "name": "CPUHeadroom",
 "capacity": {
 "max": 8610,
 "min": 0,
 "avg": 1602.22,
 "total": 14420
 },
 "units": "VM",
 "values": {
 "max": 8595,
 "min": 0,

```

```

 "avg":1571.44,
 "total":14143
 },
 "value":1571.44
 },
 {
 "name":"CPUExhaustion",
 "units":"Day",
 "values":{
 "max":2147483650,
 "min":0,
 "avg":2147483650,
 "total":2147483650
 },
 "value":2147483650
 },
 ...
]

```

## Getting a Filtered List of Statistics for a Market

*Referenced Endpoint:* [Stats](#)

Gets a list of statistics about the specified market, filtered by the criteria in the `StatPeriodApiInputDTO` you must pass as an inputDTO to this request. For more information about creating a filtered list of statistics, including example POSTs and responses, see *Filtered Statistics* in the [Statistics Endpoint](#).

**Example:**

**Example Input:**

**Response:** An array of `StatSnapshotApiDTO`s representing the filtered statistics for the market.

## Getting the Supply Chain for a Market

*Referenced Endpoint:* [Supply Chains](#)

Using one or more of the following parameters, you can filter the supply chain that is returned by this request:

- `market_Uuid`  
The UUID of the market whose supply chain will be returned
- `types`  
The entity types whose supply chain will be returned. Choose from:
  - APPLICATION
  - APPLICATION\_SERVER
  - BUSINESS\_ACCOUNT
  - BUSINESS\_UNIT
  - CHASSIS
  - CLOUD\_SERVICE
  - CONTAINER
  - CONTAINER\_POD
  - DATABASE
  - DATABASE\_SERVER
  - DATACENTER

- DISK\_ARRAY
- DPOD
- IO\_MODULE
- LOAD\_BALANCER
- LOGICAL\_POOL
- NETWORK
- PHYSICAL\_MACHINE
- STORAGE
- STORAGE\_CONTROLLER
- SWITCH
- VIRTUAL\_APPLICATION
- VIRTUAL\_DATACENTER
- VIRTUAL\_MACHINE
- VPOD
- entity\_states
 

Only return entities in the specified states. Choose from:

  - ACTIVE:
 

Entity is actively consuming resources.
  - EVACUATED:
 

Entity is ready to suspend and will not participate in the market, but will not generate a SUSPEND action.

**Note:** This state replaces the SUSPEND state, and requires additional configuration to enable. Turbonomic does not recommend enabling this state.
  - FAILOVER:
 

This entity is reserved for failover and will not participate in the market.
  - IDLE:
 

Entity is powered off and not actively consuming market resources.
  - LAUNCH:
 

Entity is starting.
  - MAINTENANCE:
 

Entity is in maintenance mode.
  - NOT\_MONITORED:
 

Entity is not currently monitored by Turbonomic.
  - QUEUED:
 

Entity is in the process of changing from one state to another.

**Note:** This state is internal to Turbonomic and will not appear in the UI, but may appear in an API response if the call is executed at the precise moment the entity is changing state.
  - RESOURCE\_ALLOCATION:
 

Cloud entity is being created.
  - RESOURCE\_RELEASE:
 

Cloud Entity is being deleted.
  - SUSPEND:

Entity is suspended or the guest OS is in sleep, standby, or suspended state. The entity is not removed from the environment, but will not be used in market decisions. Based on market conditions, suspended entities can be started, re-adding them to the market.

– SUSPEND\_PENDING:

Entity is making preparations to suspend, or is the next entity of its type to suspend.

– TERMINATE\_PENDING:

Entity is in the process of being deleted.

– UNKNOWN:

Turbonomic is unable to obtain the current state of the entity.

This occurs when the entity is reported to the hypervisor as having a state of DISCONNECTED, ORPHANED, or INVALID.

- environment\_type

Only return actions that affect entities in the specified environment type. [ CLOUD, HYBRID, ONPREM ].

- detail\_type

The level of detail returned for each item in the returned supply chain. [ compact, entity, aspects ].

- health

Return a small summary of the health of the returned supply chain.

**Example:** GET [https://10.10.10.10/api/v3/markets/Market/supplychains?types=VirtualMachine&entity\\_states=RESOURCE\\_ALLOCATION&entity\\_states=ACTIVE&environment\\_type=ONPREM&detail\\_type=aspects&health=true](https://10.10.10.10/api/v3/markets/Market/supplychains?types=VirtualMachine&entity_states=RESOURCE_ALLOCATION&entity_states=ACTIVE&environment_type=ONPREM&detail_type=aspects&health=true)

**Response:** A SupplychainApiDTO representing the entities meeting your criteria. In the above example, a supply chain including Virtual Machines in the ACTIVE or RESOURCE\_ALLOCATION states, in an ONPREM environment, and a healthcheck will be included at the end of the request.

```
{
 "seMap":{
 "VirtualMachine":{
 "depth":-1,
 "entitiesCount":335,
 "healthSummary":{
 "Major":6,
 "Minor":148,
 "Normal":181
 },
 "stateSummary":{
 "ACTIVE":335
 },
 "connectedProviderTypes":[
 "PhysicalMachine",
 "Storage",
 "VirtualDataCenter"
],
 "connectedConsumerTypes":[
 "Application"
],
 "instances":{
 "421edc7a-761d-9a79-9378-7140caecd50f":{
 "links":[
 {
 "rel":"self",
 "href":"https://10.10.10.10/api/v3/entities/421edc7a-761d-9a79-9378-7140caecd50f?include_aspects=false"
 }
],
 "uuid":"421edc7a-761d-9a79-9378-7140caecd50f",
 "displayName":"vm-max-test-netapp",
 "className":"VirtualMachine",
 "priceIndex":1.0070138,

```

```

 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "_8c98wBwnEem2J-ksdl1CgA",
 "displayName": "vsphere-dc7.dev.mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "vm-281",
 "aspects": {
 "virtualMachineAspect": {
 "os": "SUSE Linux Enterprise 11 (64-bit)",
 "connectedNetworks": [
 {
 "uuid": "7d6d67a50cfe1979797de159220841a37a55a50c",
 "displayName": "VM network"
 }
]
 },
 "numVCPU": 1,
 "ebsOptimized": false
 },
 "environmentType": "ONPREM"
 },
 ...
}

```

## Markets Endpoint Tips

### Unplaced Entities

When running a plan in Turbonomic, you can add workload or remove infrastructure – if the remaining resources are insufficient for the workload, entities may be "unplaced". Turbonomic provides an array of the unplaced workloads.

**Note:** An example will be forthcoming.

### Finding a Previously Run Plan Market

While the Turbonomic UI makes a plan look like one item, it is not. A plan comprises two parts:

- A scenario, which represents the configuration options for the plan. For more information about plan configuration, see [Scenarios](#).
- The plan market that is created when a scenario is run – this contains the actions, notifications, etc. that make up the state of the market after the plan.

The simplest way to discover a plan market is to execute the GET `https://10.10.10.10/api/v3/markets` and disregard the main market. The main market DTO entry will be similar to the following:

```

{
 "uuid": "777777",
 "displayName": "Market",
 "className": "Market",
 "state": "RUNNING",
 "stateProgress": 100,
 "unplacedEntities": false,
 "environmentType": "HYBRID"
}

```

The other markets that the request returns are plan markets. Plan market DTOs are similar to:

```

{
 "uuid": "214172431889696",
 "displayName": "Add Workload 2",
 "className": "Market",
 "state": "SUCCEEDED"
}

```

```

 "stateProgress":100,
 "scenario":{
 "uuid":"214172431885856",
 "displayName":"Add Workload 2",
 "owners":[
 {
 "uuid":"2993567556912",
 "showSharedUserSC":false
 }
]
 }
 }
}

```

The outer-most `uuid` and `displayName` parameters are the UUID and display name of plan market. The `state` field shows the results of the plan run, where `SUCCEEDED` means the plan finished successfully, and `FAILED` means the plan did not finish running- usually because it was cancelled by the user.

Within the plan itself, the `scenario` parameter includes information about the scenario used on the market. Note the inner `uuid` and `displayName` are the UUID and display name of the scenario, which you can use to see all markets created using this scenario.

Finally, iterating over the return of the `/markets` call above, and looking at each market and comparing the UUID of the scenario (in this case, `214172431889696`), you can see how many times the scenario was run, and also each market containing the results of that run.

## Notifications Endpoint

Notifications are system and market messages from your Turbonomic instance.

This endpoint contains references to start and end times for certain operations. For more information, including valid time formats, see [Time in the Turbonomic API](#)

Using the notifications endpoint, you can:

- Get all notifications
- Get a filtered list of notifications
- Get related actions for a specified notification

## Notification Categories

Each notification from your Turbonomic instance belongs to a particular category of the following:

- Discovery
- Monitoring
- Control
- Mediation
- Healthcheck
- InterAppliance
- MarketAction
- MarketProblem
- Presentation
- Deploy
- Security

## Notifications Requests

### Getting Notifications

Gets a list of system and market level notifications. For a list of notifications, the API returns an array of LogEntryApiDTOs. Each LogEntryApiDTO will contain details about the notification, including the time, state, severity, and importance of the notification. To get a single notification, include the UUID of the notification in the request.

This request takes the following parameters:

- **start\_time**  
The start time for a historic range of notifications.
- **end\_time**  
The end time for a historic range of notifications.
- **category**  
The category of notifications to return. [Discovery, Monitoring, Control, Mediation, Healthcheck, InterAppliance, MarketAction, MarketProblem, Presentation, Deploy, Security]
- **cursor**  
When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.
- **limit**  
The maximum number of items to return. Must be a positive integer. If not set, the server will provide a default (20).
- **order\_by**  
The field to order the results by. This value can be any field found in the response.
- **ascending**  
Default: true. When false, results will be in descending order.

#### Examples:

- **All Notifications (no parameters):** GET <https://10.10.10.10/api/v3/notifications>
- **All Notifications (parameters):** GET [https://10.10.10.10/api/v3/notifications?start\\_time=1237056032&end\\_time=1238056032&limit=100&order\\_by=name&ascending=true](https://10.10.10.10/api/v3/notifications?start_time=1237056032&end_time=1238056032&limit=100&order_by=name&ascending=true)
- **Single Notification:** GET [https://10.10.10.10/api/v3/notifications/\\_5wcf4J5WEeinZ-6Uu4soUw](https://10.10.10.10/api/v3/notifications/_5wcf4J5WEeinZ-6Uu4soUw)

#### Response:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/notifications/_5wcf4J5WEeinZ-6Uu4soUw"
 }
],
 "uuid": "_5wcf4J5WEeinZ-6Uu4soUw",
 "logActionTime": 1534095671518,
 "targetSE": "Hyper-V:hp-cluster.corp.mycorp.com",
 "category": "Discovery",
 "subCategory": "Validation",
 "description": "Validation of target hp-cluster.corp.mycorp.com failed",
 "shortDescription": "Validation Failed",
 "state": "NOTIFY",
 "severity": "MAJOR",
 "importance": 0
 }
]
```

```

 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/notifications/_sQ7xUJ5XEeinZ-6Uu4soUw"
 }
],
 "uuid": "_sQ7xUJ5XEeinZ-6Uu4soUw",
 "logActionTime": 1534096010470,
 "targetSE": "Hyper-V:hp-cluster.corp.mycorp.com",
 "category": "Discovery",
 "subCategory": "Discovery",
 "description": "Could not get domain for the 'hp-cluster.corp.mycorp.com' target",
 "shortDescription": "Discovery failed",
 "state": "NOTIFY",
 "severity": "MAJOR",
 "importance": 0
 },
 ...
]

```

## Parking Endpoint

Turbonomic supports 'parking' actions for cloud resources. These actions *stop* your cloud resources for a given period of time to help you reduce your cloud expenses, and then *start* these resources later when you need them. You can enforce parking actions on demand or according to a schedule.

Using the Parking endpoint of the Turbonomic API, you can:

- Get a list of parkable entities
- Enforce parking actions on demand

To enforce parking actions according to a schedule, use `/schedules/timespans/` requests in the Schedules endpoint. With these requests, you can configure parking schedules and then attach the schedules to parkable entities. For details, see [Parking Schedules Requests](#).

When parking schedules have been attached to entities, you can use the Parking endpoint to pause and resume schedules as needed.

## Parking Requests

### Pagination for Parking Requests

When you request parking entities, the API uses pagination to return subsets of the data. You should always use pagination when available to make sure you get your data in manageable chunks. To paginate returned data, the `limit` and `cursor` parameters control how the APIs paginate their data.

**Example:** POST `https://10.10.10.10/api/v3/parking/entities?cursor=10&limit=5&order_by=DISPLAY_NAME`

This example returns the 11th through 16th entities ordered by display name.

For general information about pagination, see [Pagination in the Turbonomic API](#).

### Getting Parkable Entities

Provides a list of entities that can be stopped or started.

**Example:** POST `https://10.10.10.10/api/v3/parking/entities`

Results may be sorted by one of the following: `DISPLAY_NAME`, `TYPE`, `STATE`, `ACCOUNT_NAME`, `PROVIDER`, `REGION_NAME`, `INSTANCE_TYPE` or `COST`. The default sort order is ascending but it may be reversed by including `ascending=false` in the query string.

**Example:** POST `https://10.10.10.10/api/v3/parking/entities?order_by=DISPLAY_NAME&ascending=true`



## Getting a Filtered List of Parkable Entities

The simplest form of filtering is to provide one of the following top-level keys:

| Field       | Possible Values                                                                                     | Example                                          |
|-------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------|
| providers   | AWS, AZURE, GCP                                                                                     | <code>{"providers": ["AWS", "GCP"]}</code>       |
| status      | STOPPED, RUNNING, STARTING, STOPPING                                                                | <code>{"status": ["RUNNING", "STARTING"]}</code> |
| entityTypes | VirtualMachine, DatabaseServer<br><b>Note:</b><br>DatabaseServer only applies to AWS RDS instances. | <code>{"entityTypes": ["VirtualMachine"]}</code> |

**Example:** POST `https://10.10.10.10/api/v3/parking/entities`

Example input that fetches all running VMs in all of your AWS targets:

```
{"entityTypes": ["VirtualMachine"], "status":["RUNNING"], "providers": ["AWS"]}
```

**Example:** POST `https://10.10.10.10/api/v3/parking/entities`

Example input, showing the same query with the addition of the field `logicalOperator`. The `logicalOperator` value can either be AND or OR, and will control whether the supplied filters are all required or are optional. For example, this request now fetches all entities that are a virtual machine, or are running, or are provided by an AWS target:

```
{"entityTypes": ["VirtualMachine"], "status": "RUNNING", "providers": ["AWS"], "logicalOperator": "OR"}
```

### Advanced Filtering

More powerful filtering is available by using `criteriaList`. This object supports additional fields and provides more control over your filter expressions, including regular expression filtering. Each element in a `criteriaList` contains the following keys: `expVal`, `expType`, and `filterType`.

| filterType   | expType                   | Constraints                                                                                              |
|--------------|---------------------------|----------------------------------------------------------------------------------------------------------|
| displayName  | EQ, NEQ, RXEQ, RXNEQ      |                                                                                                          |
| accountName  | EQ, NEQ, RXEQ, RXNEQ      |                                                                                                          |
| regionName   | EQ, NEQ, RXEQ, RXNEQ      |                                                                                                          |
| instanceType | EQ, NEQ, RXEQ, RXNEQ      |                                                                                                          |
| scheduleName | EQ, NEQ, RXEQ, RXNEQ      |                                                                                                          |
| tags         | EQ, NEQ, RXEQ, RXNEQ      | Discovered tags containing pipes   may produce unexpected results when using an expType of RXEQ or RXNEQ |
| cost         | EQ, NEQ, GT, LT, GTE, LTE |                                                                                                          |

**Example:** POST <https://10.10.10.10/api/v3/parking/entities>

Example input that fetches all entities within the "Engineering" account:

```
{
 "criteriaList": [
 {
 "expVal": "Engineering",
 "expType": "EQ",
 "filterType": "accountName"
 }
]
}
```

Advanced filtering may also take advantage of `logicalOperator`, if desired.

**Example:** POST <https://10.10.10.10/api/v3/parking/entities>

Example input that returns all entities in either the "Engineering" or "Test" accounts:

```
{
 "criteriaList": [
 {
 "expVal": "Engineering",
 "expType": "EQ",
 "filterType": "accountName"
 },
 {
 "expVal": "Test",
 "expType": "EQ",
 "filterType": "accountName"
 }
]
 "logicalOperator": "OR"
}
```

## Executing Parking Actions

You may start or stop parkable entities on demand by submitting a manual action request. Manual actions are executed immediately.

### Note:

Manual actions will not be accepted if the entities are controlled by a schedule. For these entities, you must first pause schedule enforcement.

**Example:** POST <https://10.10.10.10/api/v3/parking/bulkActionRequest>

Example input that generates actions to start the provided entities:

```
{"action": "START", "entityUuids": ["74913973295604"]}
```

**Constraints:** action can either be START or STOP.

### Response:

This API responds with one element for each `entityUuid` provided and indicates the resulting state.

```
[{"state": "STARTING", "entityUUID": "74913973295604"}]
```

If an action could not be generated for a particular `entityUuid`, the response will contain an error message:

```
[{"error": "unable to toggle the entity", "entityUUID": "74913973295605"}]
```

If an action is requested for multiple entities, a response may contain a partial success. Check the response body to understand whether the action succeeded for a particular entity.

```
[{"state": "STARTING", "entityUUID": "74913973295604"}, {"error": "unable to toggle the entity", "entityUUID": "74913973295605"}]
```

## Pausing and Resuming Parking Schedules

Parking schedules must first be configured and attached to entities using `/schedules/timespans` requests. For details, see [Parking Schedules Requests](#).

After attaching schedules to entities, you can pause or resume schedules using the Parking endpoint.

### Pausing Parking Schedules

There are times when pausing schedule enforcement is useful. The most common use is when you need to pause an environment during non-typical work hours or when an unanticipated maintenance window requires automation outside Turbonomic to start and stop workloads. Like [attaching and detaching schedules](#), you may pause or resume 1-N entities from schedule enforcement.

**Example:** POST `https://10.10.10.10/api/v3/parking/enforcementPauses`

Example input that pauses schedule enforcement for 1-N entities using a duration in minutes (only supported unit):

```
{"entityUuids":["74913973295604"],"durationMinutes":15}
```

Example input that pauses schedule enforcement for 1-N entities until 12:26 PM Central (UTC-5) in ISO 8601 date and time format:

```
{"entityUuids":["74913973295604"],"until":"2023-05-16T14:26:00-05:00"}
```

### Resuming Parking Schedules

**Example:** DELETE `https://10.10.10.10/api/v3/parking/enforcementPauses`

Example input that resumes schedule enforcement for 1-N entities:

```
{"entityUuids":["74913973295604"]}
```

## Policies Endpoint

Policies set business rules to control how Turbonomic analyzes resource allocation, how it displays resource status, and how it recommends or executes actions. There are two kinds of policies:

- Placement Policies

To modify workload placement decisions, Turbonomic divides its market into segments that constrain the valid placement of workloads. Turbonomic discovers placement rules that are defined by the targets in your environment, and you can create your own segments.

- Automation Policies

Turbonomic uses Automation Policies to guide its analysis and resulting actions. These policies can specify Action Automation, Orchestration, and Constraints (operational, utilization, and scaling constraints).

Turbonomic ships with default settings that we believe will give you the best results from our analysis and control. For some scopes of your environment, you might want to change these settings. For example, you might want to change action automation for that scope, or change the utilization constraints.

The hypervisors that you set as targets can include placement policies of their own. Turbonomic imports these placement policies, and you can choose to enable or disable them as you wish. By default, Turbonomic enables imported placement policies.

Using the policies endpoint, you can:

- Get API entry points for policies
- See details about a specific policy

## Policies Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The policies endpoint contains a utility request that shows the entry points for policy-related requests. These can be used to find the UUID and details relating to a specific policy.

**Example:** GET <https://10.10.10.10/api/v3/policies>

**Response:** An array of Link objects, where each object contains a URL to get all policies for a specified market, entity, or group.

```
{
 "links": [
 {
 "rel": "Market policies",
 "href": "https://10.10.10.10/api/v3/markets/{uuid}/policies"
 },
 {
 "rel": "Entity policies",
 "href": "https://10.10.10.10/api/v3/entities/{uuid}/policies"
 },
 {
 "rel": "Group policies",
 "href": "https://10.10.10.10/api/v3/groups/{uuid}/policies"
 }
],
 "enabled": false
}
```

## Policies Requests

### Getting the Details of a Specified Policy

The single policy request gets the details of a specified policy, given that policy's UUID. You may use this after getting policies that affect a certain scope, to view details of those policies.

**Example:** GET <https://10.10.10.10/api/v3/policies/104:domain-c7:10.10.111.111>

**Response:** A PolicyApiDTO describing details about the policy, including the group information for the provider of the policy.

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/policies/104:domain-c7:10.10.111.111"
 }
],
 "uuid": "104:domain-c7:10.10.111.111",
 "displayName": "GROUP-DRS-LN-rule/Cluster1/10.10.111.111",
 "type": "BIND_TO_GROUP",
 "name": "GROUP-DRS-LN-rule/Cluster1/10.10.111.111",
 "enabled": true,
 "capacity": 1000000000,
 "commodityType": "DrsSegmentationCommodity",
 "consumerGroup": {
 "uuid": "7843e134bab4fb841b9cb7dd98ef04acf1f36f7b",
 "displayName": "Buyers-Group-GROUP-DRS-vmGroup1/Cluster1/10.10.111.111",
 "className": "DiscoveredGroup",
 "groupType": "VirtualMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 "providerGroup": {
 "uuid": "597a35089bfe405291911f3191cf24234d4713d1",
 "displayName": "Sellers-Group-GROUP-DRS-hostGrp1/Cluster1/10.10.111.111",
 "className": "DiscoveredGroup",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
}
```

```
}
```

## Creating a Policy

Policies are created for a specific market- either a plan market, or the real-time market. For more information about policy creation, see the [Markets](#) endpoint.

## Price Lists Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The price list endpoint contains a sample rate card file and RateCardApiDTO. To access this sample:

1. Execute the GET `https://10.10.10.10/api/v3/pricelists` request, and locate the price list whose `displayName` begins with `Sample Azure Rate Card`.
2. Record the UUID of this rate card.
3. Execute the `https://10.10.10.10/api/v3/pricelists/{uuid_from_step_2}/export` request, and download the file that makes up the response body.

## Price Lists Requests

The price lists endpoint is not implemented at this time.

## Reservations Endpoint

Turbonomic uses reservations to save the resources you will need if you want to deploy workloads at a future date. Turbonomic calculates optimal placement for these workloads, and then it reserves whatever resources the different hosts and storage entities will need to support those workloads. For example, you can instruct Turbonomic on 01 January to reserve the resources required for 100 workloads on 07 January.

Using the reservations endpoint of the Turbonomic API, you can:

- Get a list of reservations
- Create a reservation
- Get information about a specific reservation
- Delete a reservation

### Reservation States

Reservations will be in one of the following states:

- FUTURE

The reservation is set to occur after the current date.

- INITIAL

The reservation is being created; occurs at the start time of the reservation.

- INPROGRESS

The reservation request has been sent to the market provider(s), and is awaiting results.

- INVALID

This error state occurs when a reservation is placed on an unready market (for example, when the market component is restarting) or one of the constraints or templates used in the reservation has been deleted between the creation of the reservation, and the start date.

- PLACEMENT\_FAILED

The reservation has completed, but there were insufficient resources to place all entities.

- RESERVED

The reservation has completed and was able to obtain all needed resources.

- UNFULFILLED

The reservation has been added to the database, but has not yet been sent to the market.

## Reservations in the Product

From the Workload Placement Page, you can set up reservations to save the resources you will need to deploy workloads at a future date. Turbonomic uses its intelligent workload management to calculate optimal placement for these workloads, and then it reserves whatever resources the different hosts and storage entities will need to support those workloads.

When creating a reservation in the product, you will select the VM template, any constraints (scope, policy, network), then give your reservation a name, virtual machine count, and start and end dates.

## Reservations Requests

### Getting Reservations

For a list of reservations, the API returns a list of DemandReservationApiDTOs. Each DemandReservationApiDTO will contain the entity information for the reserved instances, the count to be deployed, and current status. To get the DTO of a particular reservation, include the UUID in the request.

This request takes the following parameters:

#### Status

Returns only reservations in the specified status. [ PLACEMENT\_SUCCEEDED, PLACEMENT\_FAILED, UNFULFILLED, RESERVED, DEPLOY\_FAILED, DEPLOY\_SUCCEEDED, FUTURE ]

#### Examples:

- All reservations with a status of PLACEMENT\_SUCCEEDED:

```
GET https://10.10.10.10/api/v3/reservations?status=PLACEMENT_SUCCEEDED
```

- A single reservation:

```
GET https://10.10.10.10/api/v3/reservations/_TszrkcwHEei7A9D0cc4f1g
```

#### Response:

```
[
 {
 "uuid": "215446856141616",
 "displayName": "MyReservation",
 "count": 1,
 "status": "RESERVED",
 "reserveDateTime": "2022-11-01T13:28:00Z",
 "expireDateTime": "2022-12-01T14:28:00Z",
 "constraintInfos": [
 {
 "uuid": "74635127126226",
 "constraintType": "DATA_CENTER"
 }
],
 "reservationDeployed": false,
 "mode": "NO_GROUPING",
 "grouping": "NONE",
 "demandEntities": [
 {
 "template": {
 "uuid": "215364307544240",
 "displayName": "headroomVM",
 "className": "VirtualMachineProfile"
 }
 }
]
 }
]
```



in placing the reserved VMs then the `placements` field will contain descriptions of the host and storage providers for the two VMs.

```
{
 "uuid": "215447101807136",
 "displayName": "MyReservation1",
 "count": 2,
 "status": "FUTURE",
 "reserveDateTime": "2022-11-01T21:03:22Z",
 "expireDateTime": "2022-12-01T21:03:22Z",
 "constraintInfos": [
 {
 "uuid": "285815619396128",
 "constraintType": "POLICY"
 }
],
 "reservationDeployed": false,
 "mode": "NO_GROUPING",
 "grouping": "NONE",
 "demandEntities": [
 {
 "template": {
 "uuid": "215364307544240",
 "displayName": "headroomVM",
 "className": "VirtualMachineProfile"
 },
 "placements": {}
 },
 {
 "template": {
 "uuid": "215364307544240",
 "displayName": "headroomVM",
 "className": "VirtualMachineProfile"
 },
 "placements": {}
 }
]
}
```

## Deleting a Reservation

Deletes the specified reservation and removes all reserved resources associated with that reservation. This will also remove any pending deployment related to the deleted reservation.

**Example:** DELETE `https://10.10.10.10/api/v3/reservations/_i-FLwdLiEeiuY7KDiLoLWg`

**Response:** A successful deletion request returns a Response Code of 200.

## Reservations Endpoint Tips

### Customizing a Reservation

When creating a reservation, you should be familiar with [placement policies](#), [templates](#), scope constraints, and network constraints.

#### Note:

Use network constraint to limit workload placement to the networks that Turbonomic discovered in your environment.

This request takes the following parameters of the `DemandReservationApiInputDTO`:

#### **demandName**

The display name of the reservation — If you do not specify names for the reserved workloads, they will all get this as the root name, with a numeric suffix. For example, `MyReservation1`, `MyReservation2`, etc.

#### **action**

The action type: This is always `RESERVATION`.



**reserveDateTime**

The time to calculate the workload placement and create the reservation — This time cannot be earlier than the time that you POST the reservation to the API.

**expireDateTime**

When to cancel the reservation. At this time, Turbonomic will delete the reservation. To take advantage of the reserved resources, you should deploy the actual VMs to their reserved locations before this time.

**parameters**

A DemandReservationParametersDTO, which is itself composed of a PlacementParametersDTO and DeploymentParametersDTO.

In order to pass a completed DemandReservationParametersDTO, you will create a PlacementParametersDTO with the following parameters:

**count**

The number of workloads in this reservation

**templateID**

The UUID of the template used for creating the workload required for the placement

**entityNames**

Name of the placed workload, which will be affixed with a numeric suffix for reservations of more than one workload

**constraintIDs**

The UUIDs of the scope, group and any policies or network constraints that the reservation will adhere to

For example, the following POST creates a reservation of 10 VMs, scoped to a specific virtual data center with a UUID of 73365042986774 and network with a UUID of 73365048169751. Because it does not specify entityNames (an array of names for the VMs), Turbonomic will generate the VM names.

```
{
 "action": "RESERVATION",
 "demandName": "DalReservation",
 "reserveDateTime": "2020-03-11T15:44:41Z",
 "expireDateTime": "2020-04-11T15:44:41Z",
 "parameters": [
 {
 "placementParameters": {
 "count": "10",
 "templateID": "214103358655713",
 "constraintIDs": [
 "73365042986774",
 "73365048169751"
]
 }
 }
]
}
```

**Affinity Rules**

When creating reservations, you can specify affinity rules for the reserved VMs, to ensure all the VMs in a reservation deploy to the same cluster. If you do not specify affinity, the VMs deploy to any clusters in the reservation's scope. These settings are optional, but if you choose to specify affinity, you must specify compatible values for both settings.

These setting respect other constraints you might have set for the reservation. For example, if you scope the reservation to a group of clusters, with a cluster affinity rule Turbonomic will choose one cluster from that scope and try to place all the VMs on it.

To specify affinity for a reservation, you declare:

- mode (optional):

Can be one of:

- AFFINITY - Group reserved VMs in the same cluster. The `grouping` parameter must be set to `CLUSTER`. If there is no cluster that can host the requested VMs, the reservation will be `UNFULLFILLED`.
  - `NO_GROUPING` - Do not group reserved VMs in the same cluster. The `grouping` parameter must be set to `NONE`.
- `grouping` (optional):

Can be one of:

- `CLUSTER` - Group reserved VMs. The `mode` parameter must be set to `CLUSTER`.
- `NONE` - Do not group reserved VMs. The `mode` parameter must be set to `NO_GROUPING`.

```
{
 "demandName": "MyReservation1",
 "action": "RESERVATION",
 "mode": "AFFINITY",
 "grouping": "CLUSTER",
 ...
}
```

## Getting Reservation Information

To get a list of current reservations, execute the GET `https://10.10.10.10/api/v3/reservations` request. This returns a list of all active reservations. If you know the UUID of the reservation you want to retrieve, you can pass it to get data for just that reservation.

Each reservation object describes:

- Display name
- Status — can be:
  - `IN_PROGRESS` — Placement calculation is in progress
  - `PLACEMENT_SUCCEEDED` — For a new reservation, the environment has sufficient resources to place the workload; if you specified a reserve date, this will be an active reservation
  - `PLACEMENT_FAILED` — For a newly created reservation, the environment doesn't have resources to place the workload; if you specified a reserve date, this will be an unfulfilled reservation
  - `RETRYING` — Turbonomic is trying to place the workload of an unfulfilled reservation
- Time the reservation was created, and time it will expire
- A description of the reserved workload
- Statistics for the compute and storage resources the reservation sets aside

For example, this listing shows a successful reservation for one VM:

```
{
 "uuid": "_kWZHIDmEeePgeXuo0RRbw",
 "displayName": "MyReservation",
 "count": 1,
 "status": "PLACEMENT_SUCCEEDED",
 "reserveDateTime": "Thu May 04 16:27:29 UTC 2017",
 "expireDateTime": "Thu Aug 31 16:27:29 UTC 2017",
 "deployDateTime": "Thu Aug 31 16:27:29 UTC 2017",
 "reserveCount": 1,
 "demandEntities": [
 {
 "uuid": "_kWgb7TDmEeePgeXuo0RRbw",
 "displayName": "MyReservation_C0",
 "className": "VirtualMachine",
 "template": {
 "uuid": "T423f548d-cadc-e525-6df4-1f90724cf696",
 "displayName": "vsphere-dc3.dev.mycorp.com::TMP-SUSE64",
 "className": "VirtualMachineProfile"
 }
 }
],
}
```

```

"placements":{
 "computeResources":[
 {
 "stats":[
 {
 "name":"numOfCpu",
 "value":1
 },
 {
 "name":"cpuSpeed",
 "value":2603
 },
 {
 "name":"cpuConsumedFactor",
 "value":0.5
 },
 {
 "name":"memorySize",
 "value":2097152
 },
 {
 "name":"memoryConsumedFactor",
 "value":0.75
 },
 {
 "name":"ioThroughput",
 "value":0
 },
 {
 "name":"networkThroughput",
 "value":0
 }
],
 "provider":{
 "uuid":"Virtual_ESX_42381da5-12fa-1e82-2f1c-887419380d43",
 "displayName":"hp-esx21.corp.vmturbo.com",
 "className":"PhysicalMachine"
 }
 }
],
 "storageResources":[
 {
 "stats":[
 {
 "name":"diskSize",
 "value":18432.363
 },
 {
 "name":"diskIops",
 "value":0
 }
],
 "provider":{
 "uuid":"10545c15-7687ef26",
 "displayName":"QS2:ESXDC3DS1",
 "className":"Storage"
 }
 }
]
}
]
}
}

```

## Creating a Reservation

To create a reservation, you POST an input DTO that defines the reservation's:

- **action:** The action type: RESERVATION
- **demandName:** The display name of the reservation — If you do not specify names for the added workloads, this will be the root name for new VMs
- **placementParameters:**
  - **constraintIDs:** An array of UUIDs for placement policies that will affect the calculated placement
  - **count:** The number of workloads to place

- `entityNames`: An array of names for the placed VMs — The array length should equal `count`
- `geographicRedundancy`: If `true` place the workloads on unique hosts, otherwise Turbonomic can place multiple workloads on the same host
- `templateID`: The UUID of the template that you will use to place this workload — Note that the template must include a reference to the deployment profile that you specify in `deploymentParameters`
- `expireDateTime`: When to cancel a reservation if Turbonomic cannot place all the workloads by that date — for a `RESERVATION` action, Turbonomic automatically sets the deploy time to equal this time
- `reserveDateTime`: The time to calculate the workload placement and create the reservation — This time cannot be earlier than the time that you `POST` the reservation to the API

The following listing shows an input DTO that creates a reservation. It will place four workloads, and it gives specific names to each one.

```
{
 "action": "RESERVATION",
 "demandName": "MyReservation",
 "expireDateTime": "2017-10-10T12:38:17+00:00",
 "parameters": [
 {
 "deploymentParameters": {
 "deploymentProfileID": "_c9CJMDDAEeePgeXuo0RRbw"
 },
 "placementParameters": {
 "geographicRedundancy": false,
 "count": 4,
 "entityNames": [
 "foo", "bar", "baz", "bonk"
],
 "templateID": "_UKsnkJkSEeCHcOXEhZJExA"
 }
 }
],
 "reserveDateTime": "2017-05-04T18:22:12+00:00"
}
```

## Reserved Instances Endpoint

Turbonomic analysis takes advantage of AWS Reserved Instances and Azure Reserved VM Instances to calculate optimal workload placement and to arrive at the best possible costs for your deployments on the cloud.

Using the reserved instances endpoint of the Turbonomic API, you can:

- Get a single reserved instance or list of reserved instances present in your environment
- Get statistics related to the reserved instances

## Reserved Instances Requests

### Getting Reserved Instances

For a list of reserved instances, the API returns an array of `ReservedInstanceApiDTOs`. Each `ReservedInstanceApiDTO` will contain the validation status, target type and category, and input field names and values. To get a single reserved instance, include the UUID of the reserved instance in the request.

This request takes the following parameters:

`scope`

Returns only reserved instances in the specified scope. This can be the scope of a group of entities or target. For example, to see reserved instances for a single AWS target, use that target's UUID here.

**Examples:**

- All Reserved instances: GET <https://10.10.10.10/api/v3/reservedinstances>
- Reserved Instances for a Specified Scope: GET [https://10.10.10.10//api/v3/reservedinstances?scope=\\_zeCtIMAVEei2-fP3xzx1Fw](https://10.10.10.10//api/v3/reservedinstances?scope=_zeCtIMAVEei2-fP3xzx1Fw)

**Response:**

```
[
 {
 "uuid": "aws::eu-west-1::RI::a0643140-85d7-4c5e-a7ba-4ec2fc47bb76",
 "displayName": "i3.large",
 "className": "ReservedInstance",
 "template": {
 "uuid": "aws::VMPROFILE::i3.large",
 "displayName": "i3.large",
 "discovered": false,
 "family": "i3"
 },
 "platform": "LINUX",
 "type": "STANDARD",
 "location": {
 "uuid": "aws::eu-west-1::DC::eu-west-1",
 "displayName": "aws-EU (Ireland)"
 },
 "accountId": "19265476245",
 "targetId": "_zeCtIMAVEei2-fP3xzx1Fw",
 "masterAccountId": "194356375",
 "term": {
 "units": "Year",
 "values": {
 "max": 1,
 "min": 1,
 "avg": 1,
 "total": 1
 },
 "value": 1
 },
 "payment": "PARTIAL_UPFRONT",
 "onDemandPrice": {
 "units": "$/h",
 "values": {
 "max": 0.172,
 "min": 0.172,
 "avg": 0.172,
 "total": 0.172
 },
 "value": 0.172
 },
 "costPrice": {
 "units": "$/h",
 "values": {
 "max": 0.11193608,
 "min": 0.11193608,
 "avg": 0.11193608,
 "total": 0.11193608
 },
 "value": 0.11193608
 },
 "coupons": {
 "capacity": {
 "max": 16,
 "min": 16,
 "avg": 16
 },
 "units": "RICoupon",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
 },
 "totalCoupons": 0,
 "expDate": "Sat Apr 06 19:11:19 EDT 2019",
 "instanceCount": 1,
 "tenancy": "DEFAULT",
 "upFrontCost": 490,
 "actualHourlyCost": 0.0560000017285347,
 "effectiveHourlyCost": 0.11193607478789543
 }
]
```

```
},
...
]
```

## Getting Entities Covered by Reserved Instances

Given the UUID of a reserved instance, the API returns a list of entities covered by the reserved instance.

This request takes the following parameters:

reservedInstance\_Uuid

The UUID of the reserved instance.

**Example:** GET <https://10.10.10.10/api/v3/reservedinstances/1234/coveredEntities>

**Response:** An array of BaseApiDTOs describing the entities covered by the specified RI.

```
[
 {
 "uuid": "73943736161360",
 "displayName": "pyaus",
 "className": "VirtualMachine"
 },
 {
 "uuid": "73812178348464",
 "displayName": "ptaub807disabled",
 "className": "VirtualMachine"
 },
 {
 "uuid": "73746051159552",
 "displayName": "pymm",
 "className": "VirtualMachine"
 }
]
```

## Roles Requests

The roles endpoint is not implemented at this time.

## Scenarios Endpoint

To run a plan scenario, Turbonomic creates a snapshot copy of your real-time market and modifies that snapshot according to the scenario. It then uses the Economic Scheduling Engine to perform analysis on that plan market. A scenario can modify the snapshot market by changing the workload, adding or removing hardware resources, or changing or eliminating constraints such as cluster boundaries or placement policies.

As it runs a plan, Turbonomic continuously analyzes the plan market until it arrives at the optimal conditions that market can achieve. When it reaches that point, the Economic Scheduling Engine cannot find better prices for any of the resources demanded by the workload — the plan stops running, and it displays the results as the plan's desired state.

The display includes the resulting workload distribution across hosts and datastores, as well as a list of actions the plan executed to achieve the desired result.

For example, assume a scenario that adds virtual machines to a cluster. To run the plan, Turbonomic takes a snapshot of the current market, and adds the VMs to the specified cluster. Turbonomic then runs analysis on the plan market, where each entity in the supply chain shops for the resources it needs, always looking for a better price — looking for those resources from less-utilized suppliers. This analysis continues until all the resources are provided at the best possible price.

The results might show that you can add more workload to your environment, even if you reduce compute resources by suspending physical machines. The recommended actions would then indicate which hosts you can take offline, and how to distribute your virtual machines among the remaining hosts.

The settings you make in a scenario correspond to the plan settings you can make in the user interface. These include:

- Plan scope
- Changes to workload (adding, removing, or replacing VMs or containers)
- Changes to supply (adding, removing, replacing PMs or storage)
- Enable/disable placement policies and other constraints
- Changes to action modes
- Enable/disable provisioning of supply
- Enable/disable resizing of workloads

Using the scenarios endpoint of the Turbonomic API, you can:

- Get a list of scenarios on your instance
- Create a scenario
- Get a list of markets that are connected to a particular scenario

## Scenarios Requests

### Getting Scenarios

The API returns a ScenarioApiDTO including the following fields when you execute a GET `/scenarios` request:

- `uuid`

The unique identifier for this scenario.

- `displayName`

By default, Turbonomic creates a display name that matches the scenario type. You may rename the scenario after creation using a PUT request.

- `owners`

An array of user accounts that can use the scenario. Currently, this is always the user that created the scenario.

- `type`

The type of scenario. When a user creates a scenario in the GUI, this matches the given type that user chooses from the Plan Wizard.

- `scope`

An array of scope objects to specify the scope of the plan.

- `projectionDays`

An array of days to specify how far into the future to project the plan. For example, a value of `[30, 60, 120]` would return projections after one, two, and four months, respectively.

- `topologyChanges`

An array of additions or deletions to the set of entities in the plan scope.

- `loadChanges`

An array of changes to the utilization percentage for different groups of entities within the plan scope.

- `configChanges`

An array of configuration changes for different groups of entities within the plan scope.

- `changes` - DEPRECATED

An array of scenario settings that make changes to the market before running the plan.

**Example:** GET [https://10.10.10.10/api/v3/scenarios?show\\_all=false](https://10.10.10.10/api/v3/scenarios?show_all=false)

**Response:**

```
{
 "uuid": "214704587455872",
 "displayName": "Migrate to Public Cloud 1_CLOUD_MIGRATION__CONSUMPTION",
 "type": "CLOUD_MIGRATION",
 "scope": [
 {
 "uuid": "73935388327649",
 "displayName": "aws-EU (Frankfurt)",
 "className": "Region"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "migrateList": [
 {
 "projectionDay": 0,
 "source": {
 "uuid": "285041726554576",
 "displayName": "Unknown"
 },
 "destination": {
 "uuid": "73935388327649",
 "displayName": "aws-EU (Frankfurt)",
 "className": "Region",
 "state": "ACTIVE",
 "severity": "Minor",
 "discoveredBy": {
 "uuid": "73935444966640",
 "displayName": "testmachine",
 "category": "Public Cloud",
 "type": "AWS",
 "readonly": false
 },
 "severityBreakdown": {
 "MINOR": 1
 },
 "numRelatedVMs": 2,
 "environmentType": "CLOUD",
 "vendorIds": {
 "vmturboamp": "aws::eu-central-1::DC::eu-central-1"
 }
 },
 "sources": [
 {
 "uuid": "285041726554576",
 "displayName": "Unknown"
 }
],
 "destinations": [
 {
 "uuid": "73935388327649",
 "displayName": "aws-EU (Frankfurt)",
 "className": "Region",
 "state": "ACTIVE",
 "severity": "Minor",
 "discoveredBy": {
 "uuid": "73935444966640",
 "displayName": "testmachine",
 "category": "Public Cloud",
 "type": "AWS",
 "readonly": false
 },
 "severityBreakdown": {
 "MINOR": 1
 },
 "numRelatedVMs": 2,
 "environmentType": "CLOUD",
 "vendorIds": {
 "vmturboamp": "aws::eu-central-1::DC::eu-central-1"
 }
 }
],
 "destinationEntityType": "VirtualMachine",
 "removeNonMigratingWorkloads": true
 }
]
 }
}
```



```

],
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "configChanges": {
 "addPolicyList": [],
 "removePolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "riSettingList": [
 {
 "uuid": "ri.aws.preferredOfferingClass",
 "displayName": "Type",
 "value": "STANDARD",
 "valueObjectType": "String"
 },
 {
 "uuid": "ri.aws.preferredPaymentOption",
 "displayName": "Payment",
 "value": "ALL_UPFRONT",
 "valueObjectType": "String"
 },
 {
 "uuid": "ri.aws.preferredTerm",
 "displayName": "Term",
 "value": "YEARS_1",
 "valueObjectType": "String"
 }
]
 },
 "osMigrationSettingList": [
 {
 "uuid": "linuxTargetOs",
 "displayName": "linuxTargetOs",
 "value": "LINUX",
 "valueObjectType": "String"
 },
 {
 "uuid": "linuxByol",
 "displayName": "linuxByol",
 "value": "false",
 "valueObjectType": "String"
 },
 {
 "uuid": "rhelTargetOs",
 "displayName": "rhelTargetOs",
 "value": "RHEL",
 "valueObjectType": "String"
 },
 {
 "uuid": "rhelByol",
 "displayName": "rhelByol",
 "value": "false",
 "valueObjectType": "String"
 },
 {
 "uuid": "slesTargetOs",
 "displayName": "slesTargetOs",
 "value": "SLES",
 "valueObjectType": "String"
 },
 {
 "uuid": "slesByol",
 "displayName": "slesByol",
 "value": "false",
 "valueObjectType": "String"
 },
 {
 "uuid": "windowsTargetOs",
 "displayName": "windowsTargetOs",
 "value": "WINDOWS",
 "valueObjectType": "String"
 },
 {
 "uuid": "windowsByol",
 "displayName": "windowsByol",
 "value": "false",
 "valueObjectType": "String"
 },
 {
 "uuid": "selectedMigrationProfileOption",

```

```

 "displayName": "selectedMigrationProfileOption",
 "value": "MATCH_SOURCE_TO_TARGET_OS",
 "valueObjectType": "String"
 },
 {
 "uuid": "matchToSource",
 "displayName": "matchToSource",
 "value": "true",
 "valueObjectType": "String"
 }
],
"includedCoupons": {
 "includedCouponOidsList": [],
 "iswhiteList": true
},
"subscription": {
 "uuid": "73935382668544",
 "displayName": "EngineeringSample",
 "className": "BusinessAccount",
 "accountId": "323871187550",
 "master": false,
 "cloudType": "AWS",
 "riSupported": false,
 "hasRelatedTarget": false
}
}
}

```

## Creating a Scenario

In order to create a scenario, see the [“Scenario DTO Breakdown”](#) on page 996. For examples of complete ScenarioApiDTOs, see [“Scenario Configuration Examples”](#) on page 1006.

## Deleting a Scenario

Deletes the specified scenario. Note that this does not delete plan markets you create using that scenario.

**Example:** DELETE [https://10.10.10.10/api/v3/scenarios/\\_\\_xwWYAE3EeiNKPbzHdETIA](https://10.10.10.10/api/v3/scenarios/__xwWYAE3EeiNKPbzHdETIA)

**Response:** A successful deletion request returns a Response Code of 200.

## Scenario DTO Breakdown

### Scenario DTO Overview

A scenario DTO assembles configuration settings that you use to set up and run a plan. To run a plan, you will POST the scenario to a market. For most plans, this will be the real-time market. In the case of a plan-over-plan, this will be the market created from the first scenario.

Turbonomic will then run a plan based on the combination of the scenario and the market state.

These configuration settings fall into one of several groups:

- Configuration Changes (ConfigChangesApiDTO)
  - Adding and removing placement policies, removing constraints, and providing automation settings, OS migration settings, reserved instance settings, and associated business units.
- Load Changes (LoadChangesApiDTO)
  - Modifications to the current and max utilization values, and peak baseline values.
- Time-Based Topology Changes (TimeBasedTopologyChangesApiDTO)
  - Settings to include reserved VMs and periodically add historical workload to the environment.
- Topology Changes (TopologyChangesApiDTO)
  - Settings to add, remove, replace, and migrate workload, or to specify the clusters for an Alleviate Pressure plan.

## DTO Configuration Changes

The scenario configuration changes, located in the `configChanges` parameter of the `ScenarioApiDTO`, enable you to perform the following tasks:

- `addPolicyList`  
Add a placement policy to the market generated for this plan.
- `automationSettingList`  
Change one of the available automation plan settings: VM Resize, Host Provision/Suspend, and Storage Provision/Suspend.
- `osMigrationSettingList`  
For a Migrate to Cloud plan, this parameter will contain the source and destination OS platforms, and their BYOL (Bring Your Own License) status.
- `removeConstraintList`  
Remove a constraint from the market generated for this plan.
- `removePolicyList`  
Remove a placement policy from the market generated for this plan.
- `riSettingList`  
Add information about your reserved cloud instances.
- `subscription`  
Associate a business unit to the plan.

## Adding and Removing Placement Policies

*Referenced Endpoint:* [Policies](#)

When running plans, you might want to see what your environment would look like with or without certain policies enabled, or with a new policy present. You can see the effects of these changes by passing the UUIDs of current policies, or a `PolicyApiDTO` for a new policy.

### Example input (Current Policies):

```
"addPolicyList": [
 {
 "uuid": " _VnNS0CsqEemvEL3szcXU1Q",
 "uuid": "522a3317-39f5-1a97-3b53-d725927a6ae8",
 "uuid": "52672058-a93b-d0a9-bbd9-5acd33f27882"
 }
]
```

### Example input (New Policy):

```
"addPolicyList": [
 {
 "capacity": null,
 "consumerGroup": {
 "uuid": "8c17d8e79c3db85ed5d3868de622cb89187849ce",
 "displayName": "PMS_Beta DC6\\Cluster-2",
 "className": "Group",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 "enabled": true,
 "mergeGroups": null,
 "mergeType": null,
 "name": "DalTestPolMerge",
 "displayName": "DalTestPolMerge",
 "providerGroup": {
 "uuid": "91ff08749fd278925d73b61b0f0f69d79c99fc3a",
 "displayName": "PMS_Beta DC6\\Cluster-1",
 "className": "Group",
 }
 }
]
```

```

 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 "type": "MERGE",
 "uuid": ""
 }
]

```

## Setting Plan Automation Settings

Use the plan automation settings to see the effect of your scenario if VM resizes, Host Provisioning / Suspension, and Storage Provisioning / Suspension are enabled or disabled.

### Example Input:

```

"automationSettingList":[
 {
 "uuid":"resize",
 "displayName":"Resize for VMS disabled",
 "value":"false",
 "entityType":"VirtualMachine"
 },
 {
 "uuid":"provisionPM",
 "displayName":"Provision for PHYSICALMACHINES disabled",
 "value":"false",
 "entityType":"PhysicalMachine"
 },
 {
 "uuid":"suspendPM",
 "displayName":"Suspend for PHYSICALMACHINES disabled",
 "value":"false",
 "entityType":"PhysicalMachine"
 },
 {
 "uuid":"provisionDS",
 "displayName":"Provision for STS disabled",
 "value":"false",
 "entityType":"Storage"
 },
 {
 "uuid":"suspendDS",
 "displayName":"Suspend for STS enabled",
 "value":"true",
 "entityType":"Storage"
 }
]

```

Additionally, you can use these settings to adjust operational constraints, such as provisioned values for CPU and MEM:

### Example Input:

```

"automationSettingList":[
 {
 "uuid":"capacity_MemProvisioned",
 "displayName":"Mem Overprovisioned Percentage",
 "value":"9876.0",
 "defaultValue":"1000.0",
 "categories":[
 "utilizationThresholds"
],
 "valueType":"NUMERIC",
 "min":1,
 "max":1000000,
 "entityType":"PhysicalMachine",
 "sourceGroupName":"Global"
 }
]

```

## Providing Cloud Reserved Instance Information

Turbonomic enables you to upload RI pricing information specific to the scenario.

## Example Input:

```
"riSettingList": [
 {
 "uuid": "preferredTerm",
 "displayName": "Term",
 "value": "YEARS_1",
 "entityType": "YEARS_1"
 },
 {
 "uuid": "preferredPaymentOption",
 "displayName": "Payment",
 "value": "PARTIAL_UPFRONT",
 "entityType": "PARTIAL_UPFRONT"
 },
 {
 "uuid": "preferredOfferingClass",
 "displayName": "Type",
 "value": "STANDARD",
 "entityType": "STANDARD"
 },
 {
 "uuid": "preferredCoverage",
 "displayName": "Coverage",
 "value": "80",
 "entityType": "80"
 },
 {
 "uuid": "riCoverageOverride",
 "displayName": "RI Coverage Override",
 "value": "false",
 "entityType": "false"
 }
]
```

## Removing Constraints

*Referenced Endpoint:* [Entities](#)

When running plans, you might want to see what your environment would look like with or without certain constraints. For example, if a group of virtual machines was allowed to move to any cluster of your environment.

In order to remove a constraint, you must know its UUID. To get a list of constraints for this entity, execute the `/entities/ENTITY_UUID/constraints` request for the entity whose constraint you wish to remove.

```
"removeConstraintList": [
 {
 "constraintType": "ClusterCommodity",
 "projectionDay": 0,
 "target": {
 "uuid": "a6d365f4fbf03fb9a2e8cdb29e38045e31cee41b",
 "displayName": "Beta DC6_vm",
 "className": "VirtualMachine"
 }
 }
]
```

## Setting OS Migration Settings for a Migrate to Cloud Plan

When performing a Migrate to Cloud plan, you might want to use custom OS migration settings. You can specify both the OS mapping, and the BYOL (bring your own license) status of each OS in the mapping.

### Example Input:

```
"osMigrationSettingList": [
 {
 "uuid": "linuxTargetOs",
 "displayName": "linuxTargetOs",
 "value": "LINUX"
 },
 {
 "uuid": "linuxByol",

```

```

 "displayName": "linuxByol",
 "value": "true"
 },
 {
 "uuid": "rhelTargetOs",
 "displayName": "rhelTargetOs",
 "value": "RHEL"
 },
 {
 "uuid": "rhelByol",
 "displayName": "rhelByol",
 "value": "false"
 },
 {
 "uuid": "slesTargetOs",
 "displayName": "slesTargetOs",
 "value": "WINDOWS"
 },
 {
 "uuid": "slesByol",
 "displayName": "slesByol",
 "value": "true"
 },
 {
 "uuid": "windowsTargetOs",
 "displayName": "windowsTargetOs",
 "value": "WINDOWS"
 },
 {
 "uuid": "windowsByol",
 "displayName": "windowsByol",
 "value": "true"
 },
 {
 "uuid": "showMatchSourceToTargetOsOption",
 "displayName": "showMatchSourceToTargetOsOption",
 "value": "true"
 },
 {
 "uuid": "showByolOption",
 "displayName": "showByolOption",
 "value": "false"
 },
 {
 "uuid": "showCustomOsOption",
 "displayName": "showCustomOsOption",
 "value": "true"
 },
 {
 "uuid": "matchToSource",
 "displayName": "matchToSource",
 "value": "false"
 },
 {
 "uuid": "selectedMigrationProfileOption",
 "displayName": "selectedMigrationProfileOption",
 "value": "CUSTOM_OS"
 }
]

```

## Scenario DTO Load Changes

The scenario load changes, located in the `loadChanges` parameter of the `ScenarioApiDTO`, make workload utilization changes that enable you to change the following values:

- `baselineDate`  
If your plan should be run using historical data, the date of the data to be used.
- `maxUtilizationList`  
Changes the max utilization values for a group of entities in your plan.
- `peakBaselineList`  
Changes the peak baseline values for a group of entities in your plan.
- `utilizationList`

Changes the utilization values for a group of entities in your plan.

### Changing the Baseline Date

#### Example input:

```
"baselineDate": 1552622400000
```

### Changing the Utilization For a Group of Entities

#### Example input:

```
"utilizationList": [
 {
 "percentage": -12,
 "projectionDay": 0,
 "target": {
 "displayName": "Global Environment",
 "uuid": "Market"
 }
 }
]
```

### Changing the Maximum Utilization For a Group of Entities

#### Example input:

```
"maxUtilizationList": [
 {
 "maxPercentage": 82,
 "projectionDay": 0,
 "target": {
 "displayName": "All On-prem Hosts",
 "className": "Group",
 "uuid": "_PE0v-YEUEee_hYfzgV9uYg"
 }
 },
 {
 "maxPercentage": 85,
 "projectionDay": 0,
 "target": {
 "displayName": "ACM_datastore",
 "className": "Group",
 "uuid": "c31a6f99f4409700115fc61fca71e5449ad68700"
 }
 },
 {
 "maxPercentage": 77,
 "projectionDay": 0,
 "target": {
 "displayName": "Beta DC6_datastore",
 "className": "Group",
 "uuid": "5544c35ed3d7fc0d33879d3457c219d7e40fc176"
 }
 }
]
```

## Scenario DTO Time-Based Topology Changes

The scenario time-based topology changes, located in the `timebasedTopologyChanges` parameter of the `ScenarioApiDTO`, make time-dependent topology changes to your scenario, such as:

- `addHistoryVMs`

If `true`, Turbonomic will add additional workload to your scenario each month equal to the workload added to the real-time market in the previous month.

- `includeReservation`

If `true`, Turbonomic will take current reservations and reflect their resource utilization when the scenario runs.

## Adding Historical VM Usage and/or Including VM Reservations

### Example input:

```
"timebasedTopologyChanges": [
 {
 "addHistoryVMs": "true",
 "includeReservation": "true"
 }
]
```

## Scenario DTO Topology Changes

The scenario topology changes, located in the `topologyChanges` parameter of the `ScenarioApiDTO`, make topology changes to your scenario, such as:

- `addList`  
A list of entities to add to your environment as part of plan configuration.
- `migrateList`  
A list of entities to move from one set of providers to another as part of plan configuration.
- `relievePressureList`  
In a Relieve Pressure plan, the source and target clusters.
- `removeList`  
A list of entities to be removed from your environment as part of plan configuration.
- `replaceList`  
A list of entities and the templates to replace them in your environment as part of plan configuration.

## Adding Workload

Adding workload to a scenario simulates the effect of increased VM or container density in your environment. You can add multiple copies of VM groups or individual VMs.

### Example input:

```
"addList": [
 {
 "count": 3,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine"
 }
 },
 {
 "count": 10,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "421e503d-d0c0-75f9-8cce-79421924a90d",
 "displayName": "olg_test",
 "className": "VirtualMachine"
 }
 },
 {
 "count": 5,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "421e503d-d0c0-75f9-8cce-79421924a90d",
 "displayName": "olg_test",
 "className": "VirtualMachine"
 }
 }
]
```



```

 "uuid": "420f1840-297f-5a39-9543-b72046b9126f",
 "displayName": "opensuse_with_stress_template",
 "className": "VirtualMachine"
 },
 },
 },
 "count": 1,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "53523c9a4c5d67962a96f90f59456b0c3e72aca0",
 "displayName": "Development67_vm",
 "className": "VirtualMachine"
 }
}
]

```

## Migrating Workload to the Cloud

Turbonomic includes a special plan to simulate migration of on-premise workload to the cloud, or migration of workloads from one cloud to another. This plan focuses on optimizing your costs on the cloud by choosing the best templates (most adequate compute resources) and regions to host your workloads. If analysis discovers on-premise workloads that are good candidates for Reserved Instances (RIs), then it recommends migrating to those templates, and can even recommend purchasing more RI capacity. For each group of virtual machines to be migrated (source), you must provide a destination.

### Example input:

```

"migrateList": [
 {
 "projectionDay": 0,
 "source": {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "VirtualMachine",
 "severity": "Major"
 },
 "destination": {
 "uuid": "GROUP-PMsByTargetType_Azure",
 "displayName": "PMs_Azure",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "source": {
 "uuid": "fe23c8642efc5ad2ed4fa1cc502dfb179b8dc9d2",
 "displayName": "Development DC1_vm",
 "className": "VirtualMachine",
 "severity": "Major"
 },
 "destination": {
 "uuid": "GROUP-PMsByTargetType_Azure",
 "displayName": "PMs_Azure",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "source": {
 "uuid": "f0e26cb417d8143173270b7f01c7213e027cd9e5",
 "displayName": "Development DC7_vm",
 "className": "VirtualMachine",
 "severity": "Major"
 },
 "destination": {
 "uuid": "GROUP-PMsByTargetType_Azure",
 "displayName": "PMs_Azure",
 "className": "PhysicalMachine"
 }
 }
]

```

## Relieve Pressure

Including this parameter creates an Alleviate Pressure plan to find out how to migrate workloads from a stressed or hot cluster over to a cluster with more headroom.

**Example input:**

```
"relievePressureList": [
 {
 "projectionDay": 0,
 "sources": [
 {
 "uuid": "11c63ebd77647e7b9c9fb85dd2d4c77464cbeac2",
 "displayName": "DC13\\DC13-Cluster",
 "className": "Cluster"
 }
],
 "destinations": [
 {
 "uuid": "57a4026a69ba4b46ebc430cb62d5c55eeea00695",
 "displayName": "DC13\\Physical Hosts_Cluster2 - Intel",
 "className": "Cluster"
 }
]
 }
]
```

**Removing Workload**

Removing workload in a scenario configuration simulates the effect of decreased VM or container density in your environment. You can remove VM groups or individual VMs.

**Example input:**

```
"removeList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "5E4F6056-8783-45BC-9070-35CF09457E98",
 "displayName": "AH-TestHyperV-VM001",
 "className": "VirtualMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "421d62c2-964a-0f4f-32cb-ec4cec9e26db",
 "displayName": "Anshuman_MidServer",
 "className": "VirtualMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "421e048b-f66b-539c-f743-7dcc7b40943b",
 "displayName": "Gilad_XL_Storage_Testing - Pure-1",
 "className": "VirtualMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "421e9199-7159-04f2-29fc-6a501f44d1be",
 "displayName": "Gilad_XL_Storage_Testing - Pure-2",
 "className": "VirtualMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "421d9ec6-ab97-84d0-4b12-2db88ae43e2a",
 "displayName": "MS-SQLServer-3",
 "className": "VirtualMachine"
 }
 }
]
```

## Replacing Workload

When you replace workload, you select one or more VMs or containers that you want to change (target), and then you select a template to use in their place.

### Example input:

```
"replaceList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "421ef003-e152-5a25-6e03-19698340cd01",
 "displayName": "centos_imp_7",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "42001807-bc8f-b144-8ff6-eb0d0d32cc7a",
 "displayName": "cheuk-OpsMgr-172.113",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4211806a-65f2-8be1-1015-a4abe1a2ab6f",
 "displayName": "dim-dc13amd-6.2.7_01",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "42008beb-264e-b0f6-2295-f3443e4ce7d6",
 "displayName": "enOpenShiftInfra1",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "420036f6-27c3-0252-439d-e23bcff63d02",
 "displayName": "enOpenShiftInfra2",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 },
 {
 "projectionDay": 0,
```

```

 "target": {
 "uuid": "4200ba09-71eb-3084-026e-1770c23abaa4",
 "displayName": "enOpenshiftDNS",
 "className": "VirtualMachine",
 "severity": "Minor"
 },
 "template": {
 "uuid": "_wIBz8Jj4EeC6nYMiQT1jqA",
 "displayName": "Microsoft_SQL2008-small",
 "className": "VirtualMachineProfile"
 }
 }
}
]

```

## Scenario Configuration Examples

### Basic Scenario POSTs and Responses

Most scenarios consist of multiple configuration changes. However, the UI provides several starting points for custom configuration based on plan type.

Turbonomic provides the same starting point in the API via the /scenarios endpoint and the following list of POST calls and server responses:

The following examples are simplified in order to accomplish a single goal. There are many other options you might choose in conjunction with those in the following examples.

#### Add Workload

##### POST:

```

{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Add Workload- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "Market",
 "displayName": "Global Environment",
 "className": "Market"
 }
],
 "topologyChanges": {
 "addList": [
 {
 "count": 50,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine"
 }
 }
]
 },
 "migrateList": [],
 "removeList": [],
 "replaceList": [],
 "relievePressureList": []
},

```

```
"type": "ADD_WORKLOAD"
}
```

### Response:

```
{
 "uuid": "2502057162928",
 "displayName": "Add Workload- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "ADD_WORKLOAD",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "addList": [
 {
 "projectionDays": [
 0
],
 "target": {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine"
 },
 "count": 50
 }
]
 },
 "loadChanges": {},
 "configChanges": {},
 "changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
]
 },
 {
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
 },
 {
 "index": 3,
 "type": "ADDED",
 "description": "50 AJ_Test",
 "targets": [
 {
 "uuid": "421dd4ce-a4c1-231a-71e7-64fbf598bc65",
 "displayName": "AJ_Test",
 "className": "VirtualMachine",
 "severity": "Major"
 }
],
 "value": "50",
 "projectionDays": [
 0
]
 }
]
}
```

```

],
 "timebasedTopologyChanges": {}
 }
}

```

## Decommission Hosts

### POST:

```

{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [
 {
 "uuid": "provisionPM",
 "displayName": "PROVISION for PHYSICALMACHINES disabled",
 "value": "false",
 "entityType": "PhysicalMachine"
 }
],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Decommission Hosts- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "Market",
 "displayName": "Global Environment",
 "className": "Market"
 }
],
 "topologyChanges": {
 "addList": [],
 "migratelist": [],
 "removeList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 }
],
 "replaceList": [],
 "relievePressureList": []
 },
 "type": "DECOMMISSION_HOST"
}

```

## Response:

```
{
 "uuid": "2502061312640",
 "displayName": "Decommission Hosts- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "DECOMMISSION_HOST",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "removeList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine"
 }
 }
]
 },
 "loadChanges": {},
 "configChanges": {
 "automationSettingList": [
 {
 "uuid": "provisionPM",
 "displayName": "Provision",
 "value": "false",
 "defaultValue": "RECOMMEND",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
]
 }
],
 "entityType": "PhysicalMachine",
 "sourceGroupName": "Global"
 }
}
```

```

]
},
"changes": [
{
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
]
},
{
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
},
{
 "index": 3,
 "type": "REMOVED",
 "description": "dell-pe1944.corp.vmturbo.com",
 "targets": [
 {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
},
{
 "index": 4,
 "type": "REMOVED",
 "description": "dell-pe1953.corp.vmturbo.com",
 "targets": [
 {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
},
{
 "index": 5,
 "type": "REMOVED",
 "description": "dell-pe1954.corp.vmturbo.com",
 "targets": [
 {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
},
{
 "index": 6,
 "type": "SET_ACTION_SETTING",
 "description": "Provision for PhysicalMachine disabled",
 "targets": [
 {
 "uuid": "PhysicalMachine"
 }
],
 "value": "PhysicalMachine",

```



```

 "enable": false,
 "name": "provision",
 "projectionDays": [
 0
]
 }
],
"timebasedTopologyChanges": {}
}

```

## Migrate to Cloud

### POST:

```

{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Migrate to Public Cloud- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "groupType": "VirtualMachine",
 "className": "Group"
 },
 {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "groupType": "PhysicalMachine",
 "className": "Group"
 }
],
 "topologyChanges": {
 "addList": [],
 "migratelList": [
 {
 "projectionDay": 0,
 "source": {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "VirtualMachine"
 },
 "destination": {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "className": "PhysicalMachine"
 }
 }
],
 "removeList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "_nuoxwETSEemi-4wSMH00WA",
 "displayName": "All VMs In All Cloud Zones",
 "className": "VirtualMachine"
 }
 }
],
 "replaceList": [],
 "relievePressureList": []
 },
 "type": "CLOUD_MIGRATION"
}

```

## Response:

```
{
 "uuid": "2502081867536",
 "displayName": "Migrate to Public Cloud- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "CLOUD_MIGRATION",
 "scope": [
 {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "Group",
 "groupType": "VirtualMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 },
 {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "className": "Group",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "removeList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "_nuoxwETSEemi-4wSMH00WA",
 "displayName": "All VMs In All Cloud Zones",
 "className": "Group"
 }
 }
]
 },
 "migrateList": [
 {
 "projectionDay": 0,
 "source": {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "Group"
 },
 "destination": {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "className": "Group"
 }
 }
],
 "loadChanges": {},
 "configChanges": {
 "osMigrationSettingList": [
 {
 "uuid": "matchToSource",
 "value": "true"
 }
]
 },
 "changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "Group",
 "entitiesCount": 83,
 "groupType": "VirtualMachine",

```

```

 "severity": "Major",
 "logicalOperator": "AND"
 },
 {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "className": "Group",
 "entitiesCount": 78,
 "groupType": "PhysicalMachine",
 "severity": "Normal",
 "logicalOperator": "AND"
 }
]
},
{
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
},
{
 "index": 3,
 "type": "REMOVED",
 "description": "All VMs In All Cloud Zones",
 "targets": [
 {
 "uuid": "_nuoxwETSEemi-4wSMH00WA",
 "displayName": "All VMs In All Cloud Zones",
 "className": "Group",
 "entitiesCount": 154,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "logicalOperator": "AND"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
},
{
 "index": 4,
 "type": "MIGRATION",
 "description": "Migrate [ACM_vm] to All Cloud Zones",
 "targets": [
 {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "Group",
 "entitiesCount": 83,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "logicalOperator": "AND"
 },
 {
 "uuid": "_lRrTYB--EewItqBJctLGw",
 "displayName": "All Cloud Zones",
 "className": "Group",
 "entitiesCount": 78,
 "groupType": "PhysicalMachine",
 "severity": "Normal",
 "logicalOperator": "AND"
 }
],
 "projectionDays": [
 0
]
},
{
 "index": 5,
 "type": "SET_TARGET_OS",
 "targets": [
 {
 "uuid": "4656c0ba6b976d05d1a9470ee673672022e8e926",
 "displayName": "ACM_vm",
 "className": "Group",
 "entitiesCount": 83,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "logicalOperator": "AND"
 }
]
}

```

```

],
 "projectionDays": [
 0
]
 }
],
"timebasedTopologyChanges": {}
}

```

## Optimize Cloud

### POST:

```

{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [
 {
 "uuid": "resize",
 "displayName": "resize for VMS enabled",
 "value": "true",
 "entityType": "VirtualMachine"
 }
],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [
 {
 "uuid": "preferredOfferingClass",
 "displayName": "Type",
 "value": "STANDARD",
 "entityType": "STANDARD"
 },
 {
 "uuid": "preferredTerm",
 "displayName": "Term",
 "value": "YEARS_3",
 "entityType": "YEARS_3"
 },
 {
 "uuid": "preferredPaymentOption",
 "displayName": "Payment",
 "value": "ALL_UPFRONT",
 "entityType": "ALL_UPFRONT"
 },
 {
 "uuid": "preferredCoverage",
 "displayName": "Coverage",
 "value": "80",
 "entityType": "80"
 },
 {
 "uuid": "riCoverageOverride",
 "displayName": "RI Coverage Override",
 "value": "false",
 "entityType": "false"
 }
],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Optimize Cloud- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "GROUP-PMsByTargetType_AWS",
 "displayName": "PMs_AWS",
 "groupType": "PhysicalMachine",
 "className": "Group"
 }
],
 "topologyChanges": {
 "addList": [],

```

```

 "migrateList": [],
 "removeList": [],
 "replaceList": [],
 "relievePressureList": []
 },
 "type": "OPTIMIZE_CLOUD"
}

```

## Response:

```

{
 "uuid": "2502117573632",
 "displayName": "Optimize Cloud- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "OPTIMIZE_CLOUD",
 "scope": [
 {
 "uuid": "be95c99fc2a2195ac896204766a52d0e141818a6",
 "displayName": "PMS_AWS",
 "className": "Group",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {},
 "loadChanges": {},
 "configChanges": {
 "automationSettingList": [
 {
 "uuid": "resize",
 "displayName": "resize",
 "value": "true"
 }
]
 },
 "riSettingList": [
 {
 "uuid": "preferredPaymentOption",
 "displayName": "Payment",
 "value": "ALL_UPFRONT",
 "defaultValue": "ALL_UPFRONT",
 "valueType": "STRING",
 "options": [
 {
 "label": "ALL_UPFRONT",
 "value": "ALL_UPFRONT"
 },
 {
 "label": "PARTIAL_UPFRONT",
 "value": "PARTIAL_UPFRONT"
 },
 {
 "label": "NO_UPFRONT",
 "value": "NO_UPFRONT"
 }
]
 },
 {
 "sourceGroupName": "Global"
 }
],
 {
 "uuid": "preferredTerm",
 "displayName": "Term",
 "value": "YEARS_3",
 "defaultValue": "YEARS_1",
 "valueType": "STRING",
 "options": [
 {
 "label": "YEARS_1",
 "value": "YEARS_1"
 },
 {

```

```

 "label": "YEARS_3",
 "value": "YEARS_3"
 }
],
 "sourceGroupName": "Global"
},
{
 "uuid": "preferredCoverage",
 "displayName": "Coverage",
 "value": "80",
 "defaultValue": "80",
 "valueType": "NUMERIC",
 "min": 0,
 "max": 100,
 "sourceGroupName": "Global"
},
{
 "uuid": "riCoverageOverride",
 "displayName": "RI Coverage Override",
 "value": "false",
 "defaultValue": "false",
 "valueType": "BOOLEAN",
 "sourceGroupName": "Global"
},
{
 "uuid": "preferredOfferingClass",
 "displayName": "Type",
 "value": "STANDARD",
 "defaultValue": "STANDARD",
 "valueType": "STRING",
 "options": [
 {
 "label": "STANDARD",
 "value": "STANDARD"
 },
 {
 "label": "CONVERTIBLE",
 "value": "CONVERTIBLE"
 }
],
 "sourceGroupName": "Global"
}
]
},
"changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "be95c99fc2a2195ac896204766a52d0e141818a6",
 "displayName": "PMS_AWS",
 "className": "Group",
 "entitiesCount": 50,
 "groupType": "PhysicalMachine",
 "severity": "Normal",
 "logicalOperator": "AND"
 }
]
 },
 {
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
 },
 {
 "index": 3,
 "type": "ENABLED",
 "description": "Resize enabled",
 "targets": [],
 "enable": true,
 "projectionDays": [
 0
]
 },
 {
 "index": 4,
 "type": "RI_SETTINGS",
 "targets": [
 {

```

```

 "uuid": "preferredPaymentOption|ALL_UPFRONT"
 },
 {
 "uuid": "preferredTerm|YEARS_3"
 },
 {
 "uuid": "preferredCoverage|80"
 },
 {
 "uuid": "riCoverageOverride|false"
 },
 {
 "uuid": "preferredOfferingClass|STANDARD"
 }
],
 "projectionDays": [
 0
]
 },
 "timebasedTopologyChanges": {}
}

```

## On-Prem Workload Migration (Cluster to Cluster)

### POST:

```

{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "On-Prem Workload Migration 1",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "3e47ff3351f3247367c00c08fdc5bb974695c443",
 "displayName": "DC23-Datacenter\\DC23-cluster1",
 "groupType": "PhysicalMachine",
 "className": "Cluster"
 }
],
 "topologyChanges": {
 "addList": [
 {
 "count": 1,
 "projectionDays": [
 0
],
 "target": {
 "uuid": "67d24a6f265151876e87f39475fb4ee4b4170098",
 "displayName": "AvailabilitySet::ARSEN-AVS-NCUS",
 "className": "VirtualMachine"
 }
 }
],
 "migrateList": [],
 "removeList": [],
 "replaceList": [],
 "relievePressureList": []
 },
 "type": "WORKLOAD_MIGRATION"
}

```

## Response:

```
{
 "uuid": "2502063893936",
 "displayName": "On-Prem Workload Migration- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "WORKLOAD_MIGRATION",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "addList": [
 {
 "projectionDays": [
 0
],
 "target": {
 "uuid": "a6d365f4fbf03fb9a2e8cdb29e38045e31cee41b",
 "displayName": "Beta DC6_vm",
 "className": "Group"
 },
 "count": 1
 }
]
 },
 "loadChanges": {},
 "configChanges": {},
 "changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
]
 },
 {
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
 },
 {
 "index": 3,
 "type": "ADDED",
 "description": "1 Beta DC6_vm",
 "targets": [
 {
 "uuid": "a6d365f4fbf03fb9a2e8cdb29e38045e31cee41b",
 "displayName": "Beta DC6_vm",
 "className": "Group",
 "entitiesCount": 56,
 "groupType": "VirtualMachine",
 "severity": "Major",
 "logicalOperator": "AND"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
 }
],
 "timebasedTopologyChanges": {}
}
```



```
}
```

## Hardware Refresh

### POST:

```
{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Hardware Refresh- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [
 0
],
 "scope": [
 {
 "uuid": "Market",
 "displayName": "Global Environment",
 "className": "Market"
 }
],
 "topologyChanges": {
 "addList": [],
 "migratelist": [],
 "removeList": [],
 "replacelist": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 }
]
 }
}
```

```

 "projectionDay": 0,
 "target": {
 "uuid": "Virtual_ESX_4238e933-3923-8334-37fe-3aec8fbd046f",
 "displayName": "hp-esx22.dev.mycorp.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 },
 "relievePressureList": []
},
"type": "RECONFIGURE_HARDWARE"
}

```

## Response:

```

{
 "uuid": "2502070918448",
 "displayName": "Hardware Refresh- Basic",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "RECONFIGURE_HARDWARE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
],
 "projectionDays": [
 0
],
 "topologyChanges": {
 "replaceList": [
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",

```

```

 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 },
 {
 "projectionDay": 0,
 "target": {
 "uuid": "Virtual_ESX_4238e933-3923-8334-37fe-3aec8fbd046f",
 "displayName": "hp-esx22.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "template": {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
 }
]
},
"loadChanges": {},
"configChanges": {},
"changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
]
 },
 {
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
 },
 {
 "index": 3,
 "type": "REPLACED",
 "description": "Replace [dell-pe1944.corp.vmturbo.com, dell-pe1953.corp.vmturbo.com, dell-pe1954.corp.vmturbo.com, hp-esx... with HP_DL580 G7",
 "targets": [
 {
 "uuid": "4C4C4544-0042-5A10-804A-B1C04F384331",
 "displayName": "dell-pe1944.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 {
 "uuid": "4C4C4544-004A-3610-8057-C8C04F464331",
 "displayName": "dell-pe1953.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 {
 "uuid": "4C4C4544-0059-4D10-8030-B3C04F464331",
 "displayName": "dell-pe1954.corp.vmturbo.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 {
 "uuid": "Virtual_ESX_4238e933-3923-8334-37fe-3aec8fbd046f",
 "displayName": "hp-esx22.dev.mycorp.com",
 "className": "PhysicalMachine",
 "severity": "Critical"
 },
 {
 "uuid": "_juCtwJkfEeCX4rLBy_KD2g",
 "displayName": "HP_DL580 G7",
 "className": "PhysicalMachineProfile"
 }
]
 },
 {
 "value": "0",
 "projectionDays": [
 0
]
 }
]
},
],

```

```
"timebasedTopologyChanges": {}
}
```

## Alleviate Pressure

### POST:

```
{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Alleviate Pressure- Basic",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [0],
 "scope": [
 {
 "uuid": "11c63ebd77647e7b9c9fb85dd2d4c77464cbeac2",
 "displayName": "DC13\\DC13-Cluster",
 "className": "Cluster"
 },
 {
 "uuid": "57a4026a69ba4b46ebc430cb62d5c55eaaa00695",
 "displayName": "DC13\\Physical Hosts_Cluster2 - Intel",
 "className": "Cluster"
 }
],
 "topologyChanges": {
 "addList": [],
 "migrateList": [],
 "removeList": [],
 "replaceList": [],
 "relievePressureList": [
 {
 "projectionDay": 0,
 "sources": [
 {
 "uuid": "11c63ebd77647e7b9c9fb85dd2d4c77464cbeac2",
 "displayName": "DC13\\DC13-Cluster",
 "className": "Cluster"
 }
],
 "destinations": [
 {
 "uuid": "57a4026a69ba4b46ebc430cb62d5c55eaaa00695",
 "displayName": "DC13\\Physical Hosts_Cluster2 - Intel",
 "className": "Cluster"
 }
]
 }
]
 }
},
"type": "ALLEVIATE_PRESSURE"
}
```

### Response:

```
{
 "uuid": "2531404249984",
 "displayName": "On-Prem Workload Migration 1",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
}
```

```

"type": "WORKLOAD_MIGRATION",
"scope": [
 {
 "uuid": "3e47ff3351f3247367c00c08fdc5bb974695c443",
 "displayName": "DC23-Datacenter\\DC23-cluster1",
 "className": "Cluster",
 "groupType": "PhysicalMachine",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
"projectionDays": [
 0
],
"topologyChanges": {
 "addList": [
 {
 "projectionDays": [
 0
],
 "target": {
 "uuid": "67d24a6f265151876e87f39475fb4ee4b4170098",
 "displayName": "AvailabilitySet::ARSEN-AVS-NCUS",
 "className": "DiscoveredGroup"
 },
 "count": 1
 }
]
},
"loadChanges": {},
"configChanges": {},
"changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "3e47ff3351f3247367c00c08fdc5bb974695c443",
 "displayName": "DC23-Datacenter\\DC23-cluster1",
 "className": "Cluster",
 "entitiesCount": 3,
 "groupType": "PhysicalMachine",
 "severity": "Critical",
 "logicalOperator": "AND"
 }
]
 },
 {
 "index": 2,
 "type": "PROJECTION_PERIODS",
 "projectionDays": [
 0
]
 },
 {
 "index": 3,
 "type": "ADDED",
 "description": "1 AvailabilitySet::ARSEN-AVS-NCUS",
 "targets": [
 {
 "uuid": "67d24a6f265151876e87f39475fb4ee4b4170098",
 "displayName": "AvailabilitySet::ARSEN-AVS-NCUS",
 "className": "DiscoveredGroup",
 "entitiesCount": 3,
 "groupType": "VirtualMachine",
 "severity": "Critical",
 "logicalOperator": "AND"
 }
],
 "value": "1",
 "projectionDays": [
 0
]
 }
],
"timebasedTopologyChanges": {}
}

```

The result of a plan run with no custom configuration is an optimization of your current environment, respecting all current policies and constraints.

**POST:**

```
{
 "configChanges": {
 "addPolicyList": [],
 "automationSettingList": [],
 "removeConstraintList": [],
 "removePolicyList": [],
 "riSettingList": [],
 "osMigrationSettingList": [],
 "subscription": {}
 },
 "displayName": "Custom- No Config",
 "loadChanges": {
 "utilizationList": [],
 "maxUtilizationList": []
 },
 "projectionDays": [],
 "scope": [
 {
 "uuid": "Market",
 "displayName": "Global Environment",
 "className": "Market"
 }
],
 "topologyChanges": {
 "addList": [],
 "migrateList": [],
 "removeList": [],
 "replaceList": [],
 "relievePressureList": []
 },
 "type": "CUSTOM"
}
```

**Response:**

```
{
 "uuid": "2502119526800",
 "displayName": "Custom- No Config",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "CUSTOM",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
],
 "changes": [
 {
 "index": 1,
 "type": "SCOPE",
 "scope": [
 {
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market"
 }
]
 }
]
}
```

## Scenario Future Load Configuration Example

### Future Load Plans

Future load plans are a type of plan in which commodities from active VMs are taken and applied to passive or low-workload VMs, simulating the effect on the environment if these passive VMs became active. For example, you may want to investigate the effect on your environment if your disaster recovery VMs were forced to run at historical peak values. In this case, the production VMs whose commodities will be used are the active VMs, and the currently idle disaster recovery VMs are the passive VMs.

Currently, there are two future load plan types:

- **PEAKOFPEAKS**

The peak-of-peaks future load plan replicates in the passive VMs every commodity of the active VMs reaching its historical peak utilization simultaneously, and the plan result will reflect the effect on your environment.

- **CUSTOM**

The custom future load plan replicates in the passive VMs every commodity of the active VMs reaching its current utilization simultaneously, and the plan result will reflect the effect on your environment.

The setup and configuration for both future load plans differ only in the content of the type parameter- all other steps will be identical.

The settings you make in a scenario correspond to the plan settings you can make in the user interface. These include:

- Plan scope
- Changes to workload (adding, removing, or replacing VMs or containers)
- Changes to supply (adding, removing, replacing PMs or storage)
- Enable/disable placement policies and other constraints
- Changes to action modes
- Enable/disable provisioning of supply
- Enable/disable resizing of workloads

### Future Load Prerequisites

In order to run a future load plan, you must retrieve statistics for the entities to be included in the plan. Typically, this is done utilizing the POST `/markets/market_uuid/entities/stats` request, with `startDate` and `scopes` parameters.

#### Example input for statistic retrieval:

```
{
 "period": {
 "startDate": 1237056032
 },
 "scopes": [
 "4223ae0c-c99b-f6ca-c1ef-3f367bdf5528",
 "4223bb37-0090-f332-0140-cc3cce7b3735",
 "4223dc55-78f2-2cd7-9abf-b09b1be835b4",
 "42231932-83af-10d9-f0cd-26c8a08a9e18",
 "422374a3-2c64-b613-a9a7-1f900a2314b4"
]
}
```

Once the call is completed, copy the response in its entirety- or if using Swagger, click the Download button next to the response and copy the contents of the file.

## Running the Future Load Plan

After creating the ScenarioApiDTO, the response will include the uuid of the created scenario. Use this uuid along with the uuid of the market to run the plan against (typically the real-time market) in the POST `/markets/market_uuid/scenarios/scenario_uuid` request.

## Viewing the Peak of Peak Plan

After running the plan, the response will include the uuid of the created plan market, which you should note. Results can be viewed using the GET `/markets/market_uuid` request.

## Schedules Endpoint

Schedules specify a period of time during which certain events can occur. Turbonomic uses the following types of schedules:

- Calendar Schedule

A calendar schedule is a setting that sets up a window of time when an *automation policy* takes effect. This policy can execute *non-parking* actions on public cloud or on-prem entities, or change settings that affect analysis and action generation.

To configure calendar schedules, you must use `/schedules` requests. With these requests, you can:

- Get calendar schedules
- Create calendar schedules
- Edit or delete calendar schedules
- View all policies using a specific calendar schedule

**Note:** When you configure a schedule window for a resize action, to ensure Turbonomic will execute the action during the scheduled time, you must turn off the Enforce Non Disruptive Mode setting for that scheduled policy. Even if you turn the setting off for the global policy, you still must turn the setting off for your scheduled policy. Otherwise Turbonomic will not execute the resize action.

- Parking Schedule

A parking schedule is a setting that sets up a window of time when *parking* actions (stop or start) are enforced on entities in your public cloud environment.

To configure parking schedules, you must use `/schedules/timespans/` requests. With these requests, you can:

- Get parking schedules
- Create parking schedules
- Attach parking schedules to parkable entities
- Detach parking schedules from parkable entities
- Edit or delete parking schedules

## Calendar Schedules Requests

The requests listed in this topic only apply to calendar schedules used in automation policies. For details about parking schedules that can be attached to parkable entities, see [Parking Schedules Requests](#).

## Getting Calendar Schedules

Gets a list of schedules. For a list of schedules, the API returns an array of ScheduleApiDTOs. Each ScheduleApiDTO will contain details about the schedule, including the start and end time and date and recurrence. To get a single schedule, include the UUID of the notification in the request.

### Examples:

- **All Schedules:** GET `https://10.10.10.10/api/v3/schedules`



• **Single Schedule:** GET <https://10.10.10.10/api/v3/schedules/284466929277824>

**Response:**

```
[
 {
 "uuid": "284466927892512",
 "displayName": "DalSched1",
 "startTime": "2020-02-29T16:30",
 "endTime": "2020-02-29T19:30",
 "recurrence": {
 "type": "WEEKLY",
 "daysOfWeek": [
 "Mon",
 "Fri"
],
 "interval": 1
 },
 "timeZone": "America/Denver",
 "nextOccurrence": "2020-03-02T16:30:00",
 "nextOccurrenceTimestamp": 1583191800000
 },
 {
 "uuid": "284466929277824",
 "displayName": "DalSched2",
 "startTime": "2020-02-29T16:30",
 "endTime": "2020-02-29T17:45",
 "recurrence": {
 "type": "MONTHLY",
 "daysOfMonth": [
 5
],
 "interval": 3
 },
 "timeZone": "America/Denver",
 "nextOccurrence": "2020-05-05T16:30:00",
 "nextOccurrenceTimestamp": 1588717800000
 }
]
```

## Adding a Calendar Schedule

In order to create a schedule in Turbonomic, you must provide a valid ScheduleApiDTO.

**Example:** POST <https://10.10.10.10/api/v3/schedules>

Example input, showing a schedule to be run weekly on Friday between the hours of 0000 and 0400 that starts on 2019-12-20 and has no end date:

```
{
 "displayName": "DallasSched",
 "endTime": "2019-12-20T04:00",
 "startTime": "2019-12-20T00:00",
 "timeZone": "America/Denver",
 "recurrence": {
 "type": "WEEKLY",
 "interval": 1,
 "daysOfWeek": [
 "Fri"
]
 }
}
```

## Getting Policies Using a Calendar Schedule

Given a schedule ID, retrieve any policies associated with that schedule.

**Example:** GET <https://10.10.10.10/api/v3/schedules/284466929277824/settingsPolicy>

**Response:** An array of SettingspolicyApiDTOs that represent the policies using the specified schedule.

```
[
 {
 "uuid": "284466929430656",
 "displayName": "DalStorage",
 "entityType": "Storage",
 "scopes": [
 {
 "uuid": "284460958528132",
 "displayName": "HawthorneHC_datastore",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
 "settingsManagers": [
 {
 "uuid": "automationmanager",
 "displayName": "Action Mode Settings",
 "category": "Automation",
 "settings": [
 {
 "uuid": "suspend",
 "displayName": "Suspend",
 "value": "MANUAL",
 "defaultValue": "MANUAL",
 "valueType": "STRING",
 "valueObjectType": "String",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automatic",
 "value": "AUTOMATIC"
 }
]
 }
],
 "entityType": "Storage"
 }
]
 },
 {
 "schedule": {
 "uuid": "284466929277824",
 "displayName": "DalSched2",
 "startTime": "2020-02-29T16:30",
 "endTime": "2020-02-29T17:45",
 "recurrence": {
 "type": "MONTHLY",
 "daysOfMonth": [
 5
]
 },
 "interval": 3
 },
 "timeZone": "America/Denver",
 "nextOccurrence": "2020-05-05T16:30:00",
 "nextOccurrenceTimestamp": 1588717800000
 },
 {
 "disabled": false,
 "readOnly": false,
 "default": false
 }
]
```

## Calendar Schedules Tips

When creating schedules, you can set multiple timing criteria. Below are schedules with examples of advanced timing. For basic schedules, see the [Schedules Endpoint](#).

### Bi-Weekly Calendar Schedule with Specific Days

This schedule is set to run biweekly, indicated by the "type": "WEEKLY" and "interval": 2 fields. As indicated by the daysOfWeek array, this schedule is active on Monday, Wednesday, and Friday.

InputDTO:

```
{
 "displayName": "Schedule1",
 "endDate": "2020-07-10",
 "endTime": "2020-05-06T19:30",
 "startTime": "2020-05-06T15:30",
 "timeZone": "America/Denver",
 "recurrence": {
 "type": "WEEKLY",
 "interval": 2,
 "daysOfWeek": [
 "Mon",
 "Wed",
 "Fri"
]
 }
}
```

The API response for this schedule:

```
{
 "uuid": "284559431391120",
 "displayName": "DallasTest",
 "endDate": "2020-07-10",
 "startTime": "2020-05-06T15:30",
 "endTime": "2020-05-06T19:30",
 "recurrence": {
 "type": "WEEKLY",
 "daysOfWeek": [
 "Mon",
 "Wed",
 "Fri"
],
 "interval": 2
 },
 "timeZone": "America/Denver",
 "nextOccurrence": "2020-05-06T15:30:00",
 "nextOccurrenceTimestamp": 1588800600000
}
```

### Six-Month Calendar Schedule with Specific Week of the Month

This schedule is set to run every six months, indicated by the "type": "MONTHLY" and "interval": 6 fields. As indicated by the daysOfWeek array this schedule is active on Wednesday, and the "weekOfTheMonth": 2 indicates that this schedule will only run on the second week of the month.

This schedule was created on May 5th, 2020. Converting the timestamp in the API Response to a human-readable date, we see that the next run is on Wednesday, May 13, 2020 3:30:00 PM, the second Wednesday in May.

Input DTO:

```
{
 "displayName": "DallasTest2",
 "endDate": "2020-10-19",
 "endTime": "2020-05-06T19:15",
```

```

"startTime":"2020-05-06T15:30",
"timeZone":"America/Denver",
"recurrence":{
 "type":"MONTHLY",
 "interval":6,
 "daysOfWeek":[
 "Wed"
],
 "weekOfMonth":[
 2
]
}
}
}

```

The API response for this schedule:

```

{
 "uuid":"284559434037456",
 "displayName":"DallasTest2",
 "endDate":"2020-10-19",
 "startTime":"2020-05-06T15:30",
 "endTime":"2020-05-06T19:15",
 "recurrence":{
 "type":"MONTHLY",
 "daysOfWeek":[
 "Wed"
],
 "weekOfMonth":[
 2
],
 "interval":6
 },
 "timeZone":"America/Denver",
 "nextOccurrence":"2020-05-13T15:30:00",
 "nextOccurrenceTimestamp":1589405400000
}

```

## Getting Policies Attached to a Specific Calendar Schedule

In order to see what policies are using a schedule, use the `https://10.10.10.10/api/v3/schedules/{schedule_UUID}/settingsPolicy` request.

**Response:** An array of `SettingspolicyApiDTOs` that represent the policies using the specified schedule, whose details are also included.

```

[
 {
 "uuid": "284466929430656",
 "displayName": "DalStorage",
 "entityType": "Storage",
 "scopes": [
 {
 "uuid": "284460958528132",
 "displayName": "HawthorneHC_datastore",
 "isStatic": true,
 "logicalOperator": "AND"
 }
],
 "settingsManagers": [
 {
 "uuid": "automationmanager",
 "displayName": "Action Mode Settings",
 "category": "Automation",
 "settings": [
 {
 "uuid": "suspend",
 "displayName": "Suspend",
 "value": "MANUAL",
 "defaultValue": "MANUAL",
 "valueType": "STRING",
 "valueObjectType": "String",
 "options": [

```

```

 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automatic",
 "value": "AUTOMATIC"
 }
],
 "entityType": "Storage"
 }
]
},
],
"schedule": {
 "uuid": "284466929277824",
 "displayName": "DalSched2",
 "startTime": "2020-02-29T16:30",
 "endTime": "2020-02-29T17:45",
 "recurrence": {
 "type": "MONTHLY",
 "daysOfMonth": [
 5
],
 "interval": 3
 },
 "timeZone": "America/Denver",
 "nextOccurrence": "2020-05-05T16:30:00",
 "nextOccurrenceTimestamp": 1588717800000
},
"disabled": false,
"readOnly": false,
"default": false
}
]
}
]

```

## Parking Schedules Requests

The requests listed in this topic currently only apply to parking schedules and parkable entities. For details about calendar schedules used in automation policies, see [Calendar Schedules Requests](#).

### Getting Parking Schedules

Gets a list of parking schedules. For a list of schedules, the API returns an array of parking schedule metadata. To get a single parking schedule, include the parking schedule UUID in the URL.

#### Examples:

- **All Schedules:** GET <https://10.10.10.10/api/v3/schedules/timespans>
- **Single Schedule:** GET <https://10.10.10.10/api/v3/schedules/timespans/778668020368144>

#### All Schedules Response:

```

[
 {
 "uuid": "778601653004480",
 "displayName": "testing",
 "timeZone": "America/New_York"
 },
 {
 "uuid": "778603154693344",
 "displayName": "everyday of the week",
 "timeZone": "America/New_York"
 },
 {
 "uuid": "778668020368144",
 "displayName": "Sample Schedule",
 }
]

```

```

 "description": "This is a sample schedule where entities are active during normal business
hours but may undergo maintenance in the early hours of Sunday.",
 "timeZone": "America/Chicago"
 }
]

```

### Single Schedule Response:

```

{
 "uuid": "778668020368144",
 "displayName": "Sample Schedule",
 "description": "This is a sample schedule where entities are active during normal business
hours but may undergo maintenance in the early hours of Sunday.",
 "timeZone": "America/Chicago",
 "timeSpans": {
 "type": "WeekDayTimeSpansApiDTO",
 "sunday": [
 {
 "begins": "00:00",
 "ends": "01:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "02:00",
 "ends": "03:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "IGNORE"
 }
 },
 {
 "begins": "04:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
 "monday": [
 {
 "begins": "00:00",
 "ends": "06:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "07:00",
 "ends": "16:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "ON"
 }
 },
 {
 "begins": "17:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
 "tuesday": [
 {
 "begins": "00:00",
 "ends": "06:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "07:00",

```

```

 "ends": "16:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "ON"
 }
 },
 {
 "begins": "17:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
"wednesday": [
 {
 "begins": "00:00",
 "ends": "06:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "07:00",
 "ends": "16:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "ON"
 }
 },
 {
 "begins": "17:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
"thursday": [
 {
 "begins": "00:00",
 "ends": "06:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "07:00",
 "ends": "16:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "ON"
 }
 },
 {
 "begins": "17:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
"friday": [
 {
 "begins": "00:00",
 "ends": "06:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 },
 {
 "begins": "07:00",
 "ends": "16:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "ON"
 }
 }
]

```

```

 }
 },
 {
 "begins": "17:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
],
"saturday": [
 {
 "begins": "00:00",
 "ends": "23:59",
 "policy": {
 "type": "ParkingItemApiDTO",
 "state": "OFF"
 }
 }
]
}
}
}

```

## Adding a Parking Schedule

To create a parking schedule, you must provide a valid `ParkingItemApiDTO`.

**Example:** POST `https://10.10.10.10/api/v3/schedules/timespans`

Example input, showing a schedule that causes entities to stay powered on Monday through Friday from 7:00 AM to 5:00 PM. Entities are off all other times except for a two-hour window on Sunday morning between 2:00 AM and 4:00 AM where the schedule does not enforce a powered state:

```

{
 "displayName": "Sample Schedule",
 "timeZone": "America/Chicago",
 "description": "This is a sample schedule where entities are active during normal business hours but may undergo maintenance in the early hours of Sunday.",
 "timeSpans": {
 "type": "WeekDayTimeSpansApiDTO",
 "monday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "06:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "ON"},
 "begins": "07:00",
 "ends": "16:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "17:00",
 "ends": "23:59"
 }
],
 "tuesday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "06:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "ON"},
 "begins": "07:00",
 "ends": "16:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "17:00",
 "ends": "23:59"
 }
],
 "wednesday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},

```



```

 "begins": "00:00",
 "ends": "06:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "ON"},
 "begins": "07:00",
 "ends": "16:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "17:00",
 "ends": "23:59"
 }
],
"thursday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "06:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "ON"},
 "begins": "07:00",
 "ends": "16:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "17:00",
 "ends": "23:59"
 }
],
"friday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "06:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "ON"},
 "begins": "07:00",
 "ends": "16:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "17:00",
 "ends": "23:59"
 }
],
"saturday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "23:59"
 }
],
"sunday": [
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "00:00",
 "ends": "01:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "IGNORE"},
 "begins": "02:00",
 "ends": "03:59"
 },
 {
 "policy": {"type": "ParkingItemApiDTO", "state": "OFF"},
 "begins": "04:00",
 "ends": "23:59"
 }
]
}
}
}

```

### Constraints:

- displayName value must be unique.

- `timeZone` value must be consistent with the IANA Timezone Database (for example, UTC or America/New\_York).
- `begins` and `ends` values are HH:MM representations of a 24-hour clock and are considered inclusive.
- If time spans (or days of the week) are omitted, the system behaves the same as if a corresponding time span with a policy state of IGNORE exists.
- Time spans cannot overlap.

## Attaching Parking Schedules to Entities

Once a parking schedule is created and your parkable entities are discovered, you can attach a schedule to one or more entities.

**Example:** POST `https://10.10.10.10/api/v3/schedules/timespans/778668074428784/entities`

### Constraints:

- Parking schedule only.
- Entity must be a `VirtualMachine` discovered from cloud targets (AWS, Azure, or GCP), or a `DatabaseServer` (AWS RDS instance).

Example input that attaches an existing parking schedule to 1-N entities:

```
{"entityUuids":["74913973295604","74913973295605","74913973295606"]}
```

## Detaching Parking Schedules

You can detach a parking schedule from one or more entities.

**Example:** DELETE `https://10.10.10.10/api/v3/schedules/timespans/778668074428784/entities`

Example input that detaches an existing parking schedule from a 1-N entity.

```
{"entityUuids":["74913973295604","74913973295606"]}
```

**Constraints:** Parking schedule only.

## Updating a Parking Schedule

Update an existing parking schedule. Below is an example request to update a schedule to enforce a powered off state from midnight Monday to midnight Tuesday. The schedule does not enforce a state any other time.

**Example:** PUT `https://10.10.10.10/api/v3/schedules/timespans/778668020368144`

**Response:** An array of `SettingspolicyApiDTOs` that represent the policies using the specified schedule.

```
{
 "timeZone": "America/Chicago",
 "displayName": "Sample Schedule - Updated",
 "description": "Do nothing all week, except on Monday where the attached VMs should be powered off all day",
 "timeSpans": {
 "type": "WeekDayTimeSpansApiDTO",
 "monday": [
 {
 "policy": {
 "state": "OFF",
 "type": "ParkingItemApiDTO"
 },
 "begins": "00:00",
 "ends": "23:59"
 }
]
 }
}
```

```
]
 }
}
```

### Constraints:

- `displayName` value must be unique.
- `timeZone` value must be consistent with the IANA Timezone Database (for example, UTC or America/New\_York).
- `begins` and `ends` values are HH:MM representations of a 24-hour clock and are considered inclusive.
- If time spans (or days of the week) are omitted, the system behaves the same as if a corresponding time span with a policy state of IGNORE exists.
- Time spans may not overlap.

### Deleting a Parking Schedule

Remove an unattached parking schedule.

**Example:** DELETE <https://10.10.10.10/api/v3/schedules/timespans/778668020368144>

**Constraints:** Parking schedules to be deleted must not be attached to any entity.

## Search Endpoint

A search is a query performed on the entities that participate in the Turbonomic market. The most common use of search is to locate a particular object, service entity, or group of entities that share a characteristic.

Object and Entity refer to different items in your Turbonomic topology:

Using the search endpoint of the Turbonomic API, you can:

- Obtain criteria for use in detailed searching or building groups
- Get detailed information about any object in your Turbonomic environment

### Search Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The search endpoint contains a utility request that shows the various criteria that you may use to search, for each entity type in the Turbonomic supply chain, and a second that shows the options for a specified criterion. You can then use this information to build your search criteria for more detailed groups.

The available filterTypes depend on the entity making up your search:

- VirtualMachine
  - vmsByName
  - vmsByPMName
  - vmsByStorage
  - vmsByNetwork
  - vmsByApplication
  - vmsByDatabaseServer
  - vmsByDatabaseServerVersion
  - vmsByDC
  - vmsByVDC
  - vmsByDCnested
  - vmsByNumCPUs

- vmsByMem
- vmsByGuestName
- vmsByAltName
- vmsByClusterName
- vmsByDiskArrayName
- vmsByLogicalPoolName
- vmsByTag
- vmsByState
- vmsByBusinessAccountUuid
- vmsByResourceGroupUuid
- VirtualDataCenter
  - vdcsByName
  - vdcsByVDCName
  - vdcsByTag
  - vdcsByState
- PhysicalMachine
  - pmsByName
  - pmsByStorage
  - pmsByNetwork
  - pmsBySwitch
  - pmsByNumVms
  - pmsByDC
  - pmsByMem
  - pmsByNumCPUs
  - pmsByVendorName
  - pmsByCPUModel
  - pmsByModel
  - pmsByTimezone
  - pmsByClusterName
  - pmsByTag
  - pmsByState
- Storage
  - storageByName
  - storageByTag
  - storageByVMs
  - storageByDC
  - storageByPMCluster
  - storageByState
  - volumeByDeletable
- Application Component
  - appSrvsByName
  - appsByName
- WebServer

- wbSrvsByName
- BusinessApplication
  - busAppsByName
- Database
  - databaseByName
  - databaseByTag
  - databaseByBusinessAccountUuid
  - databaseByResourceGroupUuid
- DatabaseServer
  - databaseServerByName
  - databaseServerByTag
  - databaseServerByBusinessAccountUuid
  - databaseServerByEngine
  - databaseServerByEdition
  - databaseServerByVersion
- VirtualApplication
  - vappsByName
- Cluster
  - clustersByName
  - clustersByTag
- DataCenter
  - datacentersByName
  - datacentersByTag
- Group
  - groupsByName
- StorageCluster
  - storageClustersByName
- DiskArray
  - diskarrayByName
- Zone
  - zonsByName
- Region
  - regsByName
- Network
  - netsByName
- LoadBalancer
  - lbsByName
- Chassis
  - chasByName
- StorageController
  - storagecontrollerByName

- DPod
  - dpodByName
- VPod
  - vpodByName
- LogicalPool
  - logicalPoolByName
- Switch
  - switchByName
- Container
  - containersByName
  - containersByPodName
  - containersByVMName
  - containersByNamespace
  - containerSpecsByName
- ContainerPod
  - containerPodsByName
  - containerPodsByNamespace
  - containerPodsByVMName
- ServiceEntity
 

**Note:** Groups of type `ServiceEntity` are the only groups that have multiple entity types, and are used exclusively for cloud targets.

  - workloadByBusinessAccountUuid
  - workloadByResourceGroupUuid
- Workload
  - workloadByBusinessAccountUuid
  - workloadByResourceGroupUuid
  - workloadControllersByName
  - workloadControllersByNamespace
  - workloadControllersByKind
  - ResourceGroup
  - resourceGroupByName
  - resourceGroupByUuid
  - resourceGroupByBusinessAccountUuid
- BusinessAccount
  - businessAccountByName
  - businessAccountByUuid
  - subBusinessAccountOfUuid
  - businessAccountValidationStatus
  - businessAccountTargetName
  - businessAccountCloudProvider
- BillingFamily
  - billingFamilyByName

- ViewPod
  - viewPodByName
- DesktopPool
  - desktopPoolByName
  - desktopPoolByViewPod
- BusinessUser
  - businessUserByDesktopPool
  - businessUserByName
  - businessUserByViewPod
- Cloud-Specific Filters
  - computeTiersByName
  - computeTiersByMem
  - databaseTiersByName
  - databaseTiersByDBMem
  - databaseServerTiersByName
  - databaseServerTiersByVMem

**Example:** GET <https://10.10.10.10/api/v3/search/criteria>

**Response:** An array of filter criteria, separated by entity type. For example, this is the filter criteria for Virtual Machine entities:

```
"VirtualMachine": {
 "criteria": [
 {
 "inputType": "*",
 "elements": "displayName",
 "filterCategory": "property",
 "filterType": "vmsByName"
 },
 {
 "inputType": "*",
 "elements": "HostedBy:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByPMName"
 },
 {
 "inputType": "*",
 "elements": "LayeredOver:Storage:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByStorage"
 },
 {
 "inputType": "*",
 "elements": "LayeredOver:Network:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByNetwork"
 },
 {
 "inputType": "*",
 "elements": "Hosts:Application:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByApplication"
 },
 {
 "inputType": "*",
 "elements": "HostedBy:PhysicalMachine:HostedBy:DataCenter:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByDC"
 },
 {
 "inputType": "*",
 "elements": "LayeredOver:VirtualDataCenter:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByVDC"
 }
]
}
```

```

 {
 "inputType": "*",
 "elements": "AllMemberOf:VirtualDataCenter:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByDCnested"
 },
 {
 "inputType": "#",
 "elements": "ExtendedBy:VMTopoExt:numCpus",
 "filterCategory": "property",
 "filterType": "vmsByNumCPUs"
 },
 {
 "inputType": "#",
 "elements": "Commodities:VMem:capacity",
 "filterCategory": "property",
 "filterType": "vmsByMem"
 },
 {
 "inputType": "*",
 "elements": "ExtendedBy:VMTopoExt:guestName",
 "filterCategory": "property",
 "filterType": "vmsByGuestName"
 },
 {
 "inputType": "*",
 "elements": "ExtendedBy:VMTopoExt:altName",
 "filterCategory": "property",
 "filterType": "vmsByAltName"
 },
 {
 "inputType": "*",
 "elements": "HostedBy:PhysicalMachine:MemberOf:Cluster:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByClusterName"
 },
 {
 "inputType": "*",
 "elements": "LayeredOver:Storage:LayeredOver:DiskArray:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByDiskArrayName"
 },
 {
 "inputType": "*",
 "elements": "LayeredOver:Storage:LayeredOver:LogicalPool:displayName",
 "filterCategory": "entity",
 "filterType": "vmsByLogicalPoolName"
 },
 {
 "inputType": "*",
 "elements": "tags",
 "filterCategory": "property",
 "filterType": "vmsByTag"
 },
 {
 "inputType": "*",
 "elements": "state",
 "filterCategory": "property",
 "filterType": "vmsByState"
 },
 {
 "inputType": "*",
 "elements": "belongsTo:uuid",
 "filterCategory": "entity",
 "filterType": "vmsByBusinessAccountUuid"
 },
 {
 "inputType": "*",
 "elements": "MemberOf:ResourceGroup:uuid",
 "filterCategory": "entity",
 "filterType": "vmsByResourceGroupUuid"
 }
]
}

```

## Getting the Options for a Criterion



For a specific search criterion and entity type, you can get the possible values. For example, you may want to retrieve all tags for a particular group of hosts, or see the available state options for a particular group of virtual machines. This request takes the following parameters:

- `criteria_key`

The criterion to retrieve options for. Currently, Turbonomic accepts the following criteria keys: [ `state`, `tags` ]. Any other input that returns results should be considered internal or early access, and should not be relied on for scripting purposes.

- `scopes`

The UUID of the scope to retrieve the criterion values for. For example, to retrieve all tags for virtual machines in a particular cluster, put the cluster UUID here.

- `entity_type`

The entity whose state or tag options will be returned. For example, if `scopes` is passed with the UUID of a cluster, and this parameter is passed with a value of `VirtualMachine`, the request will return the options for each virtual machine in that cluster.

- `order_by`

The field to order the results by. [ `creation_date`, `name`, `risk_category`, `savings`, `severity` ].

- `ascending`

Default: `true`. When `false`, results will be in descending order.

- `environment_type`

Returns only objects in the specified environment type. [ `Cloud`, `Hybrid`, `Onprem` ]

**Example:** GET `https://10.10.10.10/api/v3/search/criteria/state/options?scopes=7c8d50fae46c09b68b00916411469770304226b4&entity_type=VirtualMachine`

**Response:** A `CriteriaOptionApiDTO` with the available options. For example, the response below indicates that all virtual machines on the cluster whose UUID was passed are either in active or idle states.

```
[
 {
 "value": "ACTIVE"
 },
 {
 "value": "IDLE"
 }
]
```

## Search Requests

### Getting Object Information by UUID

Gets information about any object in your Turbonomic environment.

`Object` and `Entity` refer to different items in your Turbonomic topology:

#### Note:

For searches, the API paginates the returned data. If you do not specify a pagination limit, then the API returns pages set to the default limit of 100. You can set a maximum of 500 for the pagination limit. If you make no settings, and your search results include more than 100 entries, the results will be paginated. Your scripts should always check for a pagination cursor when processing search results.

You can change the default and max settings in the Turbonomic `cr.yaml` file. Find the entries for:

- `apiPaginationDefaultLimit`

- `apiPaginationMaxLimit`

If you set the `objectType` parameter has a value of `Group`, the response will include objects related to an entity that perform grouping. For example, resource groups and clusters both perform grouping, and will be included in the response.

To further refine your search, include a value for the `group_type` parameter. This parameter describes the membership of the group you are searching for. For example, a search with an `objectType` of `Group` and `group_type` of `VirtualMachine` will return all groups of virtual machines in the current scope.

**Example:** `https://10.10.10.10/api/v3/search/Market`

**Response:** A DTO with information about the retrieved object. In this example, the retrieved item is the real-time Market:

```
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/markets/_0x30YUg1Ed-gHc4L513y0A"
 }
],
 "uuid": "_0x30YUg1Ed-gHc4L513y0A",
 "displayName": "Market",
 "className": "Market",
 "state": "RUNNING",
 "unplacedEntities": false,
 "environmentType": "HYBRID"
}
```

**Note:**

When making this request, you can use the `groupBy` parameter to order the results. The `/search` endpoint does not support the `utilization` value for the `groupBy` parameter. When this value is used, the API will return the results ordered by `displayName`.

## Settings Endpoint

The settings endpoint can be used to change settings in a variety of settings managers. For example, using the `actionscripmanager`, you can change action script settings, and using the `automationmanager`, you can change automation settings on your instance.

Using the settings endpoint of the Turbonomic API, you can:

- Get a list of available setting managers and settings
- Edit a setting

### List of Settings Managers

The settings managers available to be modified are:

- `actionscripmanager`  
Contains settings related to the use of action scripts.
- `automationmanager`  
Contains settings related to action modes for entities in your environment.
- `cloudsavingsmanager`  
Contains settings related to execution retention for cloud entities.
- `clusterHeadroomConfig`  
Contains settings related to the headroom plans generated by the system.
- `controlmanager`  
Contains settings related to workflows that replace native action handling for generated actions in your environment.

- `emailmanager`  
Contains settings related to SMTP email notifications.
- `hcisettingsmanager`  
Contains settings related to managing hyperconverged infrastructure.
- `marketsettingsmanager`  
Contains settings related to market analysis.
- `osmigrationmanager`  
Contains settings related to operating system migration for cloud entities.
- `persistencemanager`  
Contains settings related to statistic retention data and reporting.
- `reservedinstancemanager`  
Contains settings related to reserved instances.
- `storagesettingsmanager`  
Contains settings related to storage entity statistics and analysis.
- `telemetrymanager`  
Contains settings related to telemetry opt-in and terms of service.
- 

For more information about each settingsmanager, execute the GET `https://10.10.10.10/api/v3/settings` request.

## Settings Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The settings endpoint contains a utility request that shows each settings manager, and the settings available.

**Example:** GET `https://10.10.10.10/api/v3/settings/`

**Response:** An array of `SettingsManagerApiDTOs`, where each object is a settings manager and contains the settings for that manager.

## Settings Requests

### Getting Current Settings for a Specified Settings Manager

Gets the current settings for the specified settings manager. This request is often used to see the settings of a particular settings manager before updating.

**Example:** GET `https://10.10.10.10/api/v3/settings/automationmanager`

**Response:** A list of `SettingApiDTOs` representing the settings for the specified settings manager:

```
[
 {
 "uuid": "startVM",
 "displayName": "Start",
 "value": "RECOMMEND",
 "defaultValue": "RECOMMEND",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 }
]
 },
]
```

```

 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
},
{
 "uuid": "startPM",
 "displayName": "Start",
 "value": "RECOMMEND",
 "defaultValue": "RECOMMEND",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "PhysicalMachine",
 "sourceGroupName": "Global"
},
{
 "uuid": "startDS",
 "displayName": "Start",
 "value": "RECOMMEND",
 "defaultValue": "RECOMMEND",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "Storage",
 "sourceGroupName": "Global"
},
...
]

```

## Editing Settings

To edit a Setting, pass the new value in the body of this request. This request takes the following parameters:

- settingsManager\_uuid

The UUID of the settingsmanager whose setting will be changed.

- setting\_uuid

The setting whose value will be changed.

**Example:** PUT <https://10.10.10.10/api/v3/settings/automationmanager/moveVM>

**Example Input:** A SettingApiDTO. In this case, the value is being changed from MANUAL to RECOMMEND:

```
{
 "uuid": "moveVM",
 "displayName": "Move",
 "value": "RECOMMEND",
 "defaultValue": "MANUAL",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
}
```

**Response:** The SettingApiDTO reflecting your changes:

```
{
 "uuid": "moveVM",
 "displayName": "Move",
 "value": "RECOMMEND",
 "defaultValue": "MANUAL",
 "valueType": "STRING",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automated",
 "value": "AUTOMATIC"
 }
],
 "entityType": "VirtualMachine",
 "sourceGroupName": "Global"
}
```

## Settings Policies Endpoint

A Settings Policy is the collection of settings that apply to an entity. In Turbonomic, there is a default policy for each type of entity. These global defaults are the base settings for entities, and any settings policies that you create will specify deltas to the defaults. Note that you can edit the global policies.

When you create a custom settings policy, you assign a scope to it – one or more groups of entities you want to affect with the policy. For these policies, you specify just the settings that will be different from the defaults.

In addition to global default policies and custom policies, Turbonomic also discovers groups that need special settings, and creates policies for those groups. For example, when Turbonomic discovers Azure Availability Sets and AWS Autoscaling Groups, it generates settings policies for those groups that turn on Consistent Resizing for the group.

Using the settings policies endpoint, you can:

- Get a listing of the settings policies in your environment
- Get details for a single settings policy
- Create settings policies
- Edit settings policies
- Delete settings policies

## Settings Policies Requests

### Getting Settings Policies

This call returns an array of SettingsPolicyApiDTOs for all the settings policies in your environment. You can pass entity types to limit the listing to policies for those types, and you can limit the list to only default policies.

This request takes the following parameters:

- `only_defaults`  
If this is `true`, the call only gets the default settings policies
- `entity_type`  
One or more entity types to limit the list of settings policies

### Examples:

- Get all the settings policies: GET `https://10.10.10.10/api/v3/settingspolicies`
- Get all the default settings policies: GET `https://10.10.10.10/api/v3/settingspolicies?only_defaults=true`
- Get settings policies for VMs and Hosts: GET `https://10.10.10.10/api/v3/settingspolicies?entity_type=VirtualMachine&entity_type=PhysicalMachine`
- Get a single settings policy (pass the policy Uuid): GET `https://10.10.10.10/api/v3/settingspolicies/_aVDEwGegEemDFcIzuV5wcQ`

**Response:** An array of SettingsPolicyApiDTOs. This listing shows the settings policies for Container entities.

```
[
 {
 "uuid": "_vyMHEDnSEemHXixYkw9y2Q",
 "displayName": "Container defaults",
 "entityType": "Container",
 "settingsManagers": [
 {
 "uuid": "marketsettingsmanager",
 "displayName": "Operational Constraints",
 "category": "Analysis",
```

```

"settings": [
 {
 "uuid": "usedIncrement_Container_VMEM",
 "displayName": "Increment constant for Container VMem [MB]",
 "value": "64.0",
 "defaultValue": "64.0",
 "categories": [
 "resizeRecommendationsConstants"
],
 "valueType": "NUMERIC",
 "min": 0,
 "max": 10000000,
 "entityType": "Container",
 "sourceGroupName": "Global"
 },
 {
 "uuid": "usedIncrement_Container_VCPU",
 "displayName": "Increment constant for Container VCPU [MHz]",
 "value": "100.0",
 "defaultValue": "100.0",
 "categories": [
 "resizeRecommendationsConstants"
],
 "valueType": "NUMERIC",
 "min": 0,
 "max": 1000000,
 "entityType": "Container",
 "sourceGroupName": "Global"
 },
 {
 "uuid": "enforceConsistentResizingContainer",
 "displayName": "Consistent Resizing",
 "value": "false",
 "defaultValue": "false",
 "categories": [
 "resizeRecommendationsConstants"
],
 "valueType": "BOOLEAN",
 "entityType": "Container",
 "sourceGroupName": "Global"
 },
 {
 "uuid": "resizeTargetUtilization_Container",
 "displayName": "Resize Target Utilization",
 "value": "0.0",
 "defaultValue": "0.0",
 "valueType": "NUMERIC",
 "min": 0,
 "max": 100,
 "entityType": "Container",
 "sourceGroupName": "Global"
 }
]
...
}
]

```

## Creating a Settings Policy

*Referenced Endpoint:* [Groups](#)

To create a settings policy, you must do the following:

- Specify a policy name,
- Specify whether to enable or disable the policy in your market.
- Set the policy scope – Specify one or more groups of a given entity type. This determines the entity type for the policy.
- For the of settings you want to make, specify the appropriate settings manager, and then specify an array of setting/value pairs.
- Optionally, specify a schedule window and/or action execution schedule for the policy.

To set the policy scope, you provide an array of objects that contain group Uuids. These objects are effectively abbreviated GroupApiDTOs. Note that the groups must all be for the same type of entity. This specifies the entity type for the settings policy. Use the Groups endpoint to get the Uuid values for the groups you want in the scope. For example, to set the scope to a single group of VMs, use the following scopes object:

```
"scopes": [
 {
 "uuid": "_lSlr0GddEemDFcIzuV5wcQ"
 }
]
```

To specify the actual settings, you provide an array of SettingsManagerApiDTOs. Each one contains an array of settings that you want to make. To see all the settings for a given entity type, refer to the user interface. The setting managers you can choose from are:

- `actionscriptmanager`  
Contains settings related to the use of action scripts.
- `automationmanager`  
Contains settings related to action modes for entities in your environment.
- `cloudsavingsmanager`  
Contains settings related to execution retention for cloud entities.
- `clusterHeadroomConfig`  
Contains settings related to the headroom plans generated by the system.
- `controlmanager`  
Contains settings related to workflows that replace native action handling for generated actions in your environment.
- `emailmanager`  
Contains settings related to SMTP email notifications.
- `hcisettingsmanager`  
Contains settings related to managing hyperconverged infrastructure.
- `marketsettingsmanager`  
Contains settings related to market analysis.
- `osmigrationmanager`  
Contains settings related to operating system migration for cloud entities.
- `persistencemanager`  
Contains settings related to statistic retention data and reporting.
- `reservedinstancemanager`  
Contains settings related to reserved instances.
- `storagesettingsmanager`  
Contains settings related to storage entity statistics and analysis.
- `telemetrymanager`  
Contains settings related to telemetry opt-in and terms of service.
-



For example, to set the VM start action to AUTOMATED and to set the VCPU maximum to 16, use the following settingsManagers array:

```
"settingsManagers": [
 {
 "uuid": "automationmanager",
 "settings": [
 {
 "uuid": "startVM",
 "value": "AUTOMATIC"
 }
]
 },
 {
 "uuid": "marketsettingsmanager",
 "settings": [
 {
 "uuid": "resizeVcpuMaximum_VM",
 "value": 16
 }
]
 }
]
```

You can set up a schedule window that determines when the settings policy takes effect. For that you provide a ScheduleApiDTO for the schedule details. It includes: the start and end dates, start and end times,

- Start and end dates.
- Start and end times.
- An optional RecurrenceApiDTO for schedule recurrence. If you don't provide this you will specify a one-time schedule window.

For example, use this to specify a "perpetual" schedule window (does not expire until 2099) that starts at 12:30 am and ends at 7:30 am every Saturday and Sunday.

```
"schedule": {
 "startDate": "2019-04-28T00:30:00-04:00",
 "endDate": "2099-12-31T07:30:00-05:00",
 "startTime": "2019-04-28T00:30:00-04:00",
 "endTime": "2019-04-28T07:30:00-04:00",
 "recurrence": {
 "type": "WEEKLY",
 "daysOfWeek": [
 "Sat",
 "Sun"
]
 }
}
```

An example complete data payload for a policy with a policy-level schedule, action execution instruction, and action execution schedule is as follows:

```
{
 "disabled": false,
 "entityType": "VirtualMachine",
 "displayName": "TestPol",
 "scopes": [
 {
 "uuid": "285041619454560"
 }
],
 "settingsManagers": [
 {
 "uuid": "automationmanager",
 "displayName": "Action Mode Settings",
 "category": "Automation",
 "settings": [
 {
 "uuid": "storageMove",
 "value": "RECOMMEND",
 "valueType": "STRING",
 "valueObjectType": "String",

```

```

 "defaultValue": "RECOMMEND",
 "entityType": "VirtualMachine",
 "displayName": "Storage Move",
 "options": [
 {
 "label": "Disabled",
 "value": "DISABLED"
 },
 {
 "label": "Recommend",
 "value": "RECOMMEND"
 },
 {
 "label": "External Approval",
 "value": "EXTERNAL_APPROVAL"
 },
 {
 "label": "Manual",
 "value": "MANUAL"
 },
 {
 "label": "Automatic",
 "value": "AUTOMATIC"
 }
]
 },
 {
 "uuid": "storageMoveExecutionSchedule",
 "value": "285088800975856",
 "valueType": "LIST",
 "valueObjectType": "String",
 "entityType": "VirtualMachine",
 "displayName": "Execution window for Storage Move action"
 }
]
},
"schedule": {
 "uuid": "285088797821136",
 "displayName": "TestSched",
 "endTime": "2021-05-24T16:00",
 "startTime": "2021-05-24T14:30",
 "timeZone": "America/Chicago",
 "recurrence": {
 "type": "MONTHLY",
 "interval": 1,
 "daysOfMonth": [
 1
]
 }
}
}
}
}

```

**Example:** Using the above payload, POST <https://10.10.10.10/api/v3/settingspolicies>

**Response:** The settings policy that you created. Use this response to get the policy Uuid.

## Editing a Settings Policy

To edit a settings policy, you push a new set of data to the Settings Policies end point. The best way to do this is to copy the SettingsPolicyApiDTO that you want to edit, make changes to that DTO, and then push the DTO out to the API. To remove the policy from the scope, pass the SettingsPolicyApiDTO, and also pass `reset_defaults=true` as a parameter.

This request takes the following parameters:

- settingsPolicy\_Uuid

The Uuid of the policy you want to edit

- reset\_defaults

A boolean you can use to reset the policy to default. When `reset_defaults=true`, you remove the policy and reset that scope to default. When you use this setting, pass an empty object `{}` as the data payload.

### Examples:

- Assuming you copied a SettingsPolicyApiDTO and then made some changes to it, and pass that at the payload: PUSH `https://10.10.10.10/api/v3/settingspolicies/_2-CbQGoNEemy0oIvFZyv1A?reset_defaults=false`
- Assuming you want to remove the policy (and so you pass an empty object as the payload): PUSH `https://10.10.10.10/api/v3/settingspolicies/_2-CbQGoNEemy0oIvFZyv1A?reset_defaults=true`

**Response:** The modified SettingsPolicyApiDTO.

### Deleting a Settings Policy

Deletes the given settings policy.

**Example:** DELETE `https://10.10.10.10/api/v3/settingspolicies/_aV5ucGoTEemy0oIvFZyv1A`

**Response:** 200 for success.

## Statistics Endpoint

### Statistics Overview

Statistics are pieces of information that Turbonomic discovers while continuously analyzing your environment. The most common use of statistics is to look at a particular statistic over a specific time frame.

Using the stats endpoint of the Turbonomic API, you can:

- Get a list of stats requests in other endpoints
- Get a filtered list of statistics over a specific time frame
- Get a list of statistics for a specified entity
- Get a filtered list of statistics for a specified entity

### Statistic Timings

Statistics associated to headroom and capacity planning are written to the Turbonomic database daily, with a timestamp of midnight. Other statistics are collected and stored in the database in ten-minute increments. When retrieving groups of statistics that include both headroom and actively-monitored information, the timestamps returned will reflect the last-written date for each statistic.

### Statistics Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The stats endpoint contains a utility request that shows common entry points for statistic-related requests. These can be used to obtain statistics for particular entities or groups.

**Example:** GET `https://10.10.10.10/api/v3/stats`

**Response:** An array of Link objects, where each object contains a URL to get all statistics for a specified UUID, entity, or group.

### Statistics Requests

#### Getting Statistics for an Entity

Gets all statistics for the specified entity. This request takes the following parameters:

- entity\_uuid

The entity whose statistics will be returned.

- q

Used internally by Turbonomic.

**Note:**

When you retrieve statistics for a cluster, using the /groups endpoint will return the aggregated statistics for all applicable cluster entities. For individual stats per entity, use the /stats endpoint.

Some statistics are only valid for certain scopes. When you request given statistics for a scope that doesn't include them, the API does not return any data. For example, the numVMs statistic is valid in only the Market and Cluster scopes – the API omits numVMs data from the response when you query other scopes.

**Example:** <https://10.10.10.10/api/v3/stats/423f26a7-1132-df1c-64e2-8a040480b046>

**Response:** A list of StatSnapshotApiDTOs representing the current values of the statistics.

```
[
 {
 "displayName": "AdityaReplica",
 "date": "2019-04-24T20:09:46-04:00",
 "statistics": [
 {
 "name": "priceIndex",
 "relatedEntityType": "VirtualMachine",
 "values": {
 "max": 1.01,
 "min": 1.01,
 "avg": 1.01,
 "total": 1.01
 },
 "value": 1.01
 },
 {
 "displayName": "CPUProvisioned/AdityaReplica",
 "name": "CPUProvisioned",
 "capacity": {
 "max": 3120000,
 "min": 3120000,
 "avg": 3120000,
 "total": 3120000
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
]
 },
 {
 "units": "MHz",
 "values": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "value": 5200
 }
]
 },
 {
 "displayName": "Swapping/AdityaReplica",
 "name": "Swapping",
 "capacity": {
 "max": 40000000,
 "min": 40000000,
 "avg": 40000000,
 "total": 40000000
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
]
 },
 {
 "units": "bit/sec",
```

```

 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
 },
 {
 "displayName": "NumDisk/AdityaReplica",
 "name": "NumDisk",
 "capacity": {
 "max": 2147483650,
 "min": 2147483650,
 "avg": 2147483650,
 "total": 2147483650
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "values": {
 "max": 1,
 "min": 1,
 "avg": 1,
 "total": 1
 },
 "value": 1
 },
 ...
]
}
]

```

## Getting a Filtered List of Statistics

Gets a list of statistics filtered by a `StatPeriodApiInputDTO` that you create. For more information about the criteria that make up the input for this request, see the Swagger documentation for this request, located at [https://<Your\\_Turbonomic\\_IP>/swagger/external/index.html](https://<Your_Turbonomic_IP>/swagger/external/index.html).

**Example:** POST <https://10.10.10.10/api/v3/stats/423f26a7-1132-df1c-64e2-8a040480b046>

**Example Input:** This input will return the values for the VMem, VCPU, and VStorage statistics of the specified entity, ordered first by key, then relatedEntity, then virtualDisk.

```

{
 "statistics": [
 {
 "name": "VMem",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 },
 {
 "name": "VCPU",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 },
 {
 "name": "VStorage",
 "groupBy": [
 "key",
 "relatedEntity",
 "virtualDisk"
]
 }
]
}
]

```

```
}
```

**Response:** A list of StatSnapshotApiDTOs representing the filtered statistics for the entity:

```
[
 {
 "displayName": "AdityaReplica",
 "date": "2019-04-24T20:28:06-04:00",
 "statistics": [
 {
 "displayName": "AdityaReplica",
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 1048576
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "virtualDisk",
 "value": ""
 },
 {
 "type": "key",
 "value": null
 },
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 "relatedEntity": {
 "uuid": "5d2f7b69eb3bf506089feba37d31b0c376b02e34",
 "displayName": "GuestLoad[AdityaReplica]",
 "className": "Application"
 },
 "units": "KB",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
 },
 {
 "displayName": "AdityaReplica",
 "name": "VCPU",
 "capacity": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "virtualDisk",
 "value": ""
 },
 {
 "type": "key",
 "value": null
 },
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 "relatedEntity": {
 "uuid": "5d2f7b69eb3bf506089feba37d31b0c376b02e34",
 "displayName": "GuestLoad[AdityaReplica]",
 "className": "Application"
 },
 "units": "MHz",
 "values": {
 "max": 19,

```

```

 "min": 14.93,
 "avg": 14.93,
 "total": 14.93
 },
 "value": 14.93
 }
]
}
]

```

## Getting a Filtered List of Statistics for All Objects In a Specified Scope

For the specified scope, gets a filtered list of statistics for each entity. If you do not provide any statistics in the input, the response contains basic information about each entity. For more information about the criteria that make up the input for this request, see the Swagger documentation for this request, located at [https://<Your\\_Turbonomic\\_IP>/swagger/external/index.html](https://<Your_Turbonomic_IP>/swagger/external/index.html).

**Example:** POST <https://10.10.10.10/api/v3/stats?ascending=true>

**Example Input:** This input will return the VMem values for each entity in the real-time market from a period of five days prior to the request, to one day prior to the request.

```

{
 "scopes": [
 "Market"
],
 "period": {
 "startDate": "-5d",
 "endDate": "-1d",
 "statistics": [
 {
 "name": "VMem"
 }
]
 },
 "relatedType": "VirtualMachine"
}

```

**Response:** A list of StatSnapshotApiDTOs representing the filtered statistics for the entities in the specified scope:

```

[
 {
 "uuid": "4200829c-1272-c0a4-2d17-55f2cc0ef1b4",
 "displayName": "vm-c67a1cdd-609e-4282-809c-e0def5cd0d7a",
 "className": "VirtualMachine",
 "environmentType": "ONPREM",
 "stats": [
 {
 "displayName": "vm-c67a1cdd-609e-4282-809c-e0def5cd0d7a",
 "date": "2019-04-19T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 16777216,
 "min": 16777216,
 "avg": 16777216,
 "total": 33554432
 }
 },
 {
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 }
],
 "units": "KB",
 "values": {
 "max": 1509949.5,
 "min": 335544.3,
 "avg": 587202.56,
 "total": 1174405.1
 }
 }
]
 }
]

```

```

 },
 "value": 587202.56
 }
]
},
{
 "displayName": "vm-c67a1cdd-609e-4282-809c-e0def5cd0d7a",
 "date": "2019-04-20T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 16777216,
 "min": 16777216,
 "avg": 16777216,
 "total": 33554432
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "KB",
 "values": {
 "max": 1509949.5,
 "min": 369098.75,
 "avg": 587202.56,
 "total": 1174405.1
 },
 "value": 587202.56
 }
]
},
{
 "displayName": "vm-c67a1cdd-609e-4282-809c-e0def5cd0d7a",
 "date": "2019-04-22T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 16777216,
 "min": 16777216,
 "avg": 16777216,
 "total": 33554432
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "KB",
 "values": {
 "max": 12582912,
 "min": 0,
 "avg": 452984.84,
 "total": 905969.7
 },
 "value": 452984.84
 }
]
},
{
 "displayName": "vm-c67a1cdd-609e-4282-809c-e0def5cd0d7a",
 "date": "2019-04-23T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 16777216,
 "min": 16777216,
 "avg": 16777216,
 "total": 33554432
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],

```



```

 }
],
 "units": "KB",
 "values": {
 "max": 838860.8,
 "min": 0,
 "avg": 167772.16,
 "total": 335544.3
 },
 "value": 167772.16
 }
}
]
],
{
 "uuid": "39EBB726-41D9-4BC8-A41D-CA301854309C",
 "displayName": "Gilad_OM-38071_SMB_VM2",
 "className": "VirtualMachine",
 "environmentType": "ONPREM"
},
{
 "uuid": "97A3F420-B7AD-414D-A543-FD308AC774A6",
 "displayName": "Gilad_VMM_Test_VM_2",
 "className": "VirtualMachine",
 "environmentType": "ONPREM"
},
{
 "uuid": "4211f183-69a8-3676-3e53-0151365500ff",
 "displayName": "james-ubuntu-dc13",
 "className": "VirtualMachine",
 "environmentType": "ONPREM",
 "stats": [
 {
 "displayName": "james-ubuntu-dc13",
 "date": "2019-04-19T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "units": "KB",
 "values": {
 "max": 83886.08,
 "min": 8388.61,
 "avg": 18874.37,
 "total": 37748.73
 },
 "value": 18874.37
 }
]
 }
],
 "displayName": "james-ubuntu-dc13",
 "date": "2019-04-20T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 }
]
},
],

```

```

 "units": "KB",
 "values": {
 "max": 62914.56,
 "min": 13631.49,
 "avg": 18874.37,
 "total": 37748.73
 },
 "value": 18874.37
 },
],
 },
 {
 "displayName": "james-ubuntu-dc13",
 "date": "2019-04-21T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "units": "KB",
 "values": {
 "max": 73400.32,
 "min": 11534.34,
 "avg": 17825.79,
 "total": 35651.59
 },
 "value": 17825.79
 }
]
 },
 },
 {
 "displayName": "james-ubuntu-dc13",
 "date": "2019-04-22T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "units": "KB",
 "values": {
 "max": 83886.08,
 "min": 8388.61,
 "avg": 19922.94,
 "total": 39845.89
 },
 "value": 19922.94
 }
]
 },
 },
 {
 "displayName": "james-ubuntu-dc13",
 "date": "2019-04-23T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 }
 }
]
 }
]

```



```

 },
 "value": 37748.73
 }
]
},
{
 "displayName": "ACM-LexCorp-Control",
 "date": "2019-04-21T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 2097152,
 "min": 2097152,
 "avg": 2097152,
 "total": 4194304
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "KB",
 "values": {
 "max": 104857.6,
 "min": 20971.52,
 "avg": 35651.59,
 "total": 71303.17
 },
 "value": 35651.59
 }
]
},
{
 "displayName": "ACM-LexCorp-Control",
 "date": "2019-04-22T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 2097152,
 "min": 2097152,
 "avg": 2097152,
 "total": 4194304
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "KB",
 "values": {
 "max": 272629.75,
 "min": 20971.52,
 "avg": 73400.32,
 "total": 146800.64
 },
 "value": 73400.32
 }
]
},
{
 "displayName": "ACM-LexCorp-Control",
 "date": "2019-04-23T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 2097152,
 "min": 2097152,
 "avg": 2097152,
 "total": 4194304
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],

```

```

 }
],
 "units": "KB",
 "values": {
 "max": 251658.23,
 "min": 23068.67,
 "avg": 79691.77,
 "total": 159383.55
 },
 "value": 79691.77
}
]
}
]
},
"uuid": "420f3462-ca50-fedf-aa8f-223bd39b1852",
"displayName": "shai-redhat",
"className": "VirtualMachine",
"environmentType": "ONPREM",
"stats": [
 {
 "displayName": "shai-redhat",
 "date": "2019-04-19T00:00:00-04:00",
 "statistics": [
 {
 "name": "VMem",
 "capacity": {
 "max": 1048576,
 "min": 1048576,
 "avg": 1048576,
 "total": 2097152
 },
 "relatedEntityType": "VirtualMachine",
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "KB",
 "values": {
 "max": 20971.52,
 "min": 0,
 "avg": 1048.58,
 "total": 2097.15
 },
 "value": 1048.58
 }
]
 },
 ...
]
}
]
}
]

```

## Complex Statistic Queries

Using the POST /stats request, you can retrieve a variety of statistics. For example, this is a complex query made by the Turbonomic UI:

```

{
 "statistics": [
 {
 "name": "CPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 }
]
},

```

```

{
 "name": "Mem",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
},
{
 "name": "NetThroughput",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
},
{
 "name": "IOThroughput",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
},
{
 "name": "Q1VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
},
{
 "name": "Q2VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
},
{
 "name": "Q4VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}

```

```

],
 {
 "name": "Q8VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "name": "Q16VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "name": "Q32VCPU",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "name": "Ballooning",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "name": "Swapping",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
 },
 {
 "name": "CPUProvisioned",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",

```

```

 "value": "sold"
 }
]
},
{
 "name": "MemProvisioned",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}
},
{
 "name": "Flow",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}
},
{
 "name": "FlowAllocation",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}
},
{
 "name": "Cooling",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}
},
{
 "name": "Power",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
]
}
},
{
 "name": "Space",
 "relatedEntityType": "PhysicalMachine",
 "groupBy": [
 "key",
 "virtualDisk"
],
 "filters": [

```



```

 {
 "type": "relation",
 "value": "sold"
 }
]
}

```

The previous request is made in context of a datacenter scope, so the response includes information for each host within that datacenter. The `virtualDisk` value for the `groupBy` parameter ensures that all virtual disks appear in the response as individual line items, rather than an aggregation.

Additionally, the `"filters": [{ "type": "relation", "value": "sold" }]` found throughout the call indicates that the response includes those commodities *sold* by these hosts, rather than purchased from a lower level in the supply chain.

## Supplychains Endpoint

Turbonomic models your environment as a market of buyers and sellers. It discovers different types of entities in your environment via the targets you have configured for your installation. Discovery maps these entities to the supply chain so Turbonomic can monitor them and manage the workloads they support. For example, for a hypervisor target Turbonomic discovers VMs, the hosts and datastores that provide resources to the VMs, and the applications that use VM resources. The entities in your environment form a chain of supply and demand where some entities provide resources while others consume the supplied resources.

Using the `supplychains` endpoint, you can:

- Get a full or filtered supply chain related to an entity in your environment
- Get the statistics for the supply chain related to an entity in your environment

## Supplychains Requests

### Getting a Supply Chain for the Specified Entity

Gets the full supply chain for the specified entity. For example, if the UUID of a virtual machine is used, you will see the applications that are hosted by that virtual machine, the hosts and datastores that it consumes from, the cluster or datacenter that the hosts consume from, the disk arrays and storage controllers that the datastores consume from, and so on.

In a supply chain, depth refers to the distance in the supply chain from the entity called. For example, if the UUID of a host is used, virtual machines consuming from that host will have a depth of -1, applications consuming from those virtual machines will have a depth of -2, other hosts that are in the supply chain will have a depth of 0, and the cluster of the host will have a depth of 1.

**Example:** `GET https://10.10.10.10/api/v3/supplychains?uuids=Virtual_ESX_4238042f-7465-e92a-5c28-a7a929307247&health=false`

**Response:** A `SupplychainApiDTO` representing the supply chain of the specified entity- in this case, a host. Note that when there is a single instance of an entity, Turbonomic will return the details of that entity. In the response, you can see this with the `DPOD` and `Storage Controller` entities.

```

{
 "seMap": {
 "Application": {
 "depth": -2,
 "entitiesCount": 12,
 "stateSummary": {
 "ACTIVE": 5,
 "SUSPEND": 1,
 "IDLE": 6
 }
 }
 },

```



that grows from 1 VM > 4 ApplicationComponent > 2 Service > 1 BusinessApplication. A request starting from the VM would return the single Business Application. If you filter out BusinessApplication, it would return the two Service entities. If you filter out Service, it would return 4 ApplicationComponent entities. Note that the Swagger documentation for this call gives a list of valid filterEntityTypes.

- cursor

When making subsequent calls to retrieve additional results, this is the cursor returned by the last call to this method.

- limit

The maximum number of items to return. Must be a positive integer. If not set, the server will provide a service-specific maximum limit.

For example, in the Turbonomic user interface, you can search for the BusinessApplication entity named "App [app]". If you set the scope to that business application, then you can click in the Supply Chain to select the Namespace entity. Find the **Entity Information** chart and copy the ID value. That is the uuid for the app Namespace. If you get the leaf entities for that Namespace, you can see all the Service, ApplicationComponent, and BusinessApplication entities that support the Turbonomic application. For example, the following Service entry is for the Hyper-V target component:

```
{
 "uuid": "74626820648480",
 "displayName": "Service-app/mediation-hyperv",
 "className": "Service"
}
```

Now assume you want to use a VM group as the seeded scope, and you want to exclude any Service entities or higher in the Supply Chain:

**Example:** GET <https://10.10.168.102/api/v3/supplychains/leafEntities?uuids=74626820647952&filterEntityTypes=Service>

```
[
 {
 "uuid": "74699295824896",
 "displayName": "App-app/cost-67b7b9bb4d-b18kt/cost",
 "className": "Application Component"
 },
 {
 "uuid": "74699295824908",
 "displayName": "App-app/mediation-vmax-6cd7f4ff-nzqp6/mediation-vmax",
 "className": "Application Component"
 },
 {
 "uuid": "74699295824897",
 "displayName": "App-app/mediation-azurebilling-c585c846f-mvn17/mediation-azurebilling",
 "className": "Application Component"
 },
 ...
]
```

## Getting Statistics for a Supply Chain

Gets a list of statistics filtered by a user-created SupplyChainStatsApiInputDTO. For more information about the criteria that make up the input for this request, see the Swagger documentation for this request, located at [https://<Your\\_Turbonomic\\_IP>/swagger/external/index.html](https://<Your_Turbonomic_IP>/swagger/external/index.html).

**Example:** POST <https://10.10.168.102/api/v3/supplychains/stats>

**Example Input:** This input will return the number of applications running on the specified UUID (in this case, a Datacenter):

```
{
 "uuids": [
 "e27e5c2f85d91107ac3f26fd63d4478e808f590b"
],
}
```

```
"types": [
 "Application"
],
"environmentType": "ONPREM"
}
```

**Response:** A list of StatSnapshotApiDTOs representing the filtered statistics for the entity:

```
[
 {
 "date": "2019-05-01T17:45:42-04:00",
 "statistics": [
 {
 "name": "entities",
 "filters": [],
 "value": 25
 }
]
 }
]
```

## Tags Endpoint

### Tags Overview

Tags are metadata attached to resources that you can use to organize your environment. Tags are most often used to delineate resource groupings for technical, automation, business, or security purposes.

Turbonomic discovers tags for AWS, Azure, and vCenter resources.

Using the tags endpoint, you can:

- Get a list of available tags
- Get the members belonging to a specified tag

Additionally, you can use the tags-related requests in the entities endpoint to:

- Assign a tag to an entity
- Delete all tags or a single tag associated to an entity

### Tags Requests

#### Pagination for Tags Requests

When you request tag data, the API uses pagination to return subsets of the data. You should always use pagination when available to make sure you get your data in manageable chunks. To paginate returned data, the `limit`, `orderBy`, and `cursor` parameters control how the APIs paginate their data. If you do not specify a `limit` for the following requests, the API will paginate the data to a default limit of records:

- GET `/tags`
- GET `/tags/<tagKey>/entities`

If your scripts make these requests without setting a limit, you should be sure they check for a pagination cursor in the returned data.

For general information about pagination, see [Pagination in the Turbonomic API](#).

#### Getting Tags

For a list of tags, the API returns a list of TagApiDTOs. Each TagApiDTO will contain the key/value pair of the tag. The key can be used to see the entities belonging to the tag.

**Example:** `https://10.10.10.10/api/v3/tags`

## Response:

```
[
 {
 "key": "owner",
 "values": [
 "QA_Azure"
]
 },
 {
 "key": "director",
 "values": [
 "p-bosh",
 "bosh-init"
]
 },
 ...
]
```

## Filtering Tags by Scope, Entity, and Environment

Gets a filtered list of tags for the specified scope, entity type, and/or environment. For example, you can use this to find tags only in a particular cluster, tags that affect at least one VM, and/or limit your result to only cloud entities. This request accepts the following parameters:

- scopes

A list of UUIDs representing the scope of the request. For example, providing Market as a scope will search the entire real-time market. If you provide the UUID of a host, only entities on that host would be in the response.

- entity\_type

- APPLICATION
- APPLICATION\_SERVER
- BUSINESS\_ACCOUNT
- BUSINESS\_UNIT
- CHASSIS
- CLOUD\_SERVICE
- CONTAINER
- CONTAINER\_POD
- DATABASE
- DATABASE\_SERVER
- DATACENTER
- DISK\_ARRAY
- DPOD
- IO\_MODULE
- LOAD\_BALANCER
- LOGICAL\_POOL
- NETWORK
- PHYSICAL\_MACHINE
- STORAGE
- STORAGE\_CONTROLLER
- SWITCH
- VIRTUAL\_APPLICATION
- VIRTUAL\_DATACENTER

- VIRTUAL\_MACHINE
- VPOD
- environment\_type

Returns only tags from entities in the specified environment type. [ Cloud, Hybrid, Onprem ]

**Example:** GET [https://10.10.10.10/api/v3/tags?scopes=Market&entity\\_type=VirtualMachine&environment\\_type=ONPREM](https://10.10.10.10/api/v3/tags?scopes=Market&entity_type=VirtualMachine&environment_type=ONPREM)

**Response:** a list of VMs running on-premises in the real-time market with tags.

### Getting Entities For A Specified Tag

*Referenced Endpoint:* [Entities](#)

Gets a list of entities that have the specified tag. An entity is any device in your environment that Turbonomic manages, such as applications, physical machines, VMs, storage, and datacenters.. You can use this information to see where entities with the specified tag are located, what other tags are associated to those entities, and detailed information about each entity.

**Example:** GET <https://10.10.10.10/api/v3/tags/owner/entities>

**Response:** A list of entities that have the tag owner:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/azure::VM::375ddfb4-3127-44eb-9088-3c7645212f81?include_aspects=false"
 }
],
 "uuid": "azure::VM::375ddfb4-3127-44eb-9088-3c7645212f81",
 "displayName": "oleg-westus2-unmanaged-std",
 "className": "VirtualMachine",
 "priceIndex": 3.2109845,
 "state": "ACTIVE",
 "severity": "Normal",
 "costPrice": 0.024,
 "discoveredBy": {
 "uuid": "_GLpj4JwIEei_xJ5Ai6-rnQ",
 "displayName": "mycorp.management.core.windows.net",
 "type": "Azure"
 },
 "remoteId": "azure::VM::375ddfb4-3127-44eb-9088-3c7645212f81",
 "template": {
 "uuid": "azure::VMPROFILE::Basic_A0",
 "displayName": "Basic_A0",
 "discovered": false
 },
 "environmentType": "CLOUD",
 "tags": {
 "owner": [
 "PT_Azure"
],
 "AutoSuspend": [
 "false"
]
 }
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/azure::VM::c94e96bf-975e-47da-8c52-7ac90161cc87?include_aspects=false"
 }
],
 "uuid": "azure::VM::c94e96bf-975e-47da-8c52-7ac90161cc87",
 "displayName": "oleg-westus2-managed-avail-set-1",
 "className": "VirtualMachine",
 "priceIndex": 2.446491,
 "state": "ACTIVE",
 "severity": "Normal",
 "costPrice": 0.018,
```

```

 "discoveredBy":{
 "uuid":"_GLpj4JwIEei_xJ5Ai6-rnQ",
 "displayName":"mycorp.management.core.windows.net",
 "type":"Azure"
 },
 "remoteId":"azure::VM::c94e96bf-975e-47da-8c52-7ac90161cc87",
 "template":{
 "uuid":"azure::VMPROFILE::Standard_B1s",
 "displayName":"Standard_B1s",
 "discovered":false
 },
 "environmentType":"CLOUD",
 "tags":{
 "owner":[
 "PT_Azure"
],
 "AutoSuspend":[
 "false"
]
 }
 }
]

```

## Targets Endpoint

A target is a service that performs management functions in your virtual environment. Turbonomic uses targets to monitor workload and to execute actions in your environment.

To discover entities in your environment, you add different targets to your Turbonomic installation. Turbonomic then uses these targets to collect data from your environment.

Turbonomic uses probes to connect to targets and discover their entities. The Turbonomic GUI provides forms for users to specify different target instances. Note that each probe may require different settings, and it's possible that different versions of Turbonomic supports different sets of probes.

For complete target information, review [Target Configuration](#). You should not modify targets without a full understanding of how they interact with Turbonomic.

Using the targets endpoint of the Turbonomic API, you can:

- Get a single target or list of targets present in your environment
- Get a list of probes available in your Turbonomic installation
- Get the actions or statistics for a target
- Get the workflows discovered by a target
- Add or edit a target
- Trigger validation or rediscovery on a target
- Delete a target

## Targets Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The targets endpoint contains a utility request called specs that shows the information required for probe registry for each target available to Turbonomic, including required input fields and expected target type and category values.

### Note:

For a list of probes available to your specific Turbonomic installation, review your organization's licensing agreement.

**Example:** GET <https://10.10.10.10/api/v3/targets/specs>

**Response:** An array of TargetApiDTO objects, where each object contains an array of input fields. This example shows one entry for the JVM probe in the Application Server category.

```
[
 {
 [
 ...
]
 "category": "Application Server",
 "identifyingFields": [
 "nameOrAddress"
],
 "inputFields": [
 {
 "displayName": "Scope/Address",
 "name": "nameOrAddress",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "GROUP_SCOPE",
 "description": "The group of VMs or the address of the JVM server"
 },
 {
 "displayName": "Port Number",
 "name": "targetPort",
 "defaultValue": "1099",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "NUMERIC",
 "description": "The JMX port that connects with the JVM server"
 },
 {
 "displayName": "Username",
 "name": "username",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "STRING",
 "description": "The JVM server JMX password, enter a space if the server has no
security"
 },
 {
 "displayName": "Password",
 "name": "password",
 "isMandatory": true,
 "isSecret": true,
 "valueType": "STRING",
 "description": "The JVM server JMX port, enter a space if the server has no security"
 }
],
 "type": "JVM"
 },
 ...
]
```

## Targets Requests

Target requests handle creation, configuration, validation, rediscovery, and deletion of targets on your Turbonomic appliance.

### Pagination for Targets Requests

When you request target entities, the API uses pagination to return subsets of the data. You should always use pagination when available to make sure you get your data in manageable chunks. To paginate returned data, the `limit`, `orderBy`, and `cursor` parameters control how the APIs paginate their data.

If you do not specify a `limit` for the `GET /targets/<targetId>/entities` request, the API will paginate the data to a default limit of records. If your script makes this request without setting a `limit`, you should be sure it checks for a pagination cursor in the returned data.

For general information about pagination, see [Pagination in the Turbonomic API](#).



## Getting Targets

For a list of targets, the API returns an array of TargetApiDTO objects. Each TargetApiDTO object will contain the validation status, target type and category, and input field names and values. To get a single target, include the UUID of the target in the request.

This request takes the following parameters:

environment\_type

Returns only targets in the specified environment type. [ Cloud, Hybrid, Onprem ]

### Examples:

- All Targets: GET <https://10.10.10.10/api/v3/targets>
- Single Target: GET <https://10.10.10.10/api/v3/targets/xxwWYAE33eku7PbzHdETIA>

**Response:** A single entry in an array of targets. Notice that the password field is secret, so the response does not contain the password value.

```
[
 {
 "uuid": "xxwWYAE33eku7PbzHdETIA",
 "category": "Hypervisor",
 "type": "vCenter",
 "status": "Validated",
 "lastValidated": "2018-06-26T14:25:08-04:00",
 "inputFields": [
 {
 "displayName": "Address",
 "name": "address",
 "value": "vsphere-demo.mycorp.com",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "STRING",
 "description": "IP or address of vCenter Server",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Username",
 "name": "username",
 "value": "corp\\my.user",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "STRING",
 "description": "User name to use to connect to a target",
 "verificationRegex": ".*"
 },
 {
 "displayName": "Password",
 "name": "password",
 "isMandatory": true,
 "isSecret": true,
 "valueType": "STRING",
 "description": "Password to use to connect to a target",
 "verificationRegex": ".*"
 }
]
 }
]
```

## Adding a Target

Adds a target to your Turbonomic installation. You must pass a completed TargetApiDTO object as an inputDTO. For more information about the required fields of the TargetApiDTO for the target you want to add, execute the /targets/specs utility request.

**Example:** POST <https://10.10.10.10/api/v3/targets>

### Example inputDTO:

```
{
 "category": "Hypervisor",
 "inputFields": [
```

```

 {
 "name": "username",
 "value": "corp\my.name"
 },
 {
 "name": "password",
 "value": "Octopath"
 },
 {
 "name": "nameOrAddress",
 "value": "10.10.10.10"
 }
],
 "type": "vCenter"
}

```

## Rediscover or Validate All Targets

Manually triggers a full rediscovery or validation of all targets in your environment. This is often done to immediately see multiple changes that would normally not be reflected until a daily full discovery is performed, such as displaying updated Orchestrator workflows.

This call takes the following parameters:

- validate  
When true, validates all targets in your environment
- rediscover  
When true, performs a full rediscovery on all targets in your environment

### Examples:

- Rediscovery:  
POST <https://10.10.10.10/api/v3/targets?validate=true>
- Validation:  
POST <https://10.10.10.10/api/v3/targets?rediscover=true>

## Rediscover or Validate A Single Target

Manually triggers a full rediscovery or validation of a single target in your environment. This is often done to immediately see a change that would normally not be reflected until a daily full discovery is performed, such as displaying updated Orchestrator workflows.

This call takes the following parameters:

- validate  
When true, validates the specified target
- rediscover  
When true, performs a full rediscovery on the specified target

### Examples:

- Rediscovery: [https://10.10.10.10/api/v3/targets/\\_\\_xwWYAE33eku7PbzHdETIA?rediscover=true](https://10.10.10.10/api/v3/targets/__xwWYAE33eku7PbzHdETIA?rediscover=true)
- Validation: [https://10.10.10.10/api/v3/targets/\\_\\_xwWYAE33eku7PbzHdETIA?validate=true](https://10.10.10.10/api/v3/targets/__xwWYAE33eku7PbzHdETIA?validate=true)

**Response:** The TargetApiDTO object of the specified target:

```

{
 "uuid": "__xwWYAE3EeiNKPbzHdETIA",
 "category": "Hypervisor",
 "inputFields": [
 ...
],
}

```

```
"lastValidated": "2018-06-27T21:45:56-04:00",
"status": "Validated",
"type": "vCenter"
}
```

## Editing Target Input Fields

Edits the input fields of an existing target, and accepts the modified TargetApiDTO as input.

**Note:** You can GET the target first using the GET `/targets/specs` to see the fields required for that target. Note that you must pass all input fields in the modified TargetApiDTO object.

Example: PUT `https://10.10.10.10/api/v3/targets/_HgZQ4CdCEeaUWYEyo3vsyw`

Example input, editing the username for a single target:

```
[
 {
 "uuid": "_HgZQ4CdCEeaUWYEyo3vsyw",
 "category": "Hypervisor",
 "type": "vCenter",
 "status": "Validated",
 "lastValidated": 1474575346109,
 "inputFields": [
 {
 "displayName": "Address",
 "name": "address",
 "value": "vsphere.mycorp.com",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "STRING"
 },
 {
 "displayName": "Username",
 "name": "username",
 "value": "corp\\my.name.changed",
 "isMandatory": true,
 "isSecret": false,
 "valueType": "STRING"
 },
 {
 "displayName": "Password",
 "name": "password",
 "value": "Octopath",
 "isMandatory": true,
 "isSecret": true
 }
]
 }
]
```

## Deleting a Target

Deletes the specified target and removes all entities managed by that target. This will also remove any pending actions related to the deleted target.

**Example:** DELETE `https://10.10.10.10/api/v3/targets/xxwWYAE3EeiNKPbzHdETIA`

**Response:** A successful deletion request returns a Response Code of 200.

## Actions For a Specified Target

*Referenced Endpoint:* [Actions](#)

These requests get full or filtered lists of actions per target for use in reports, or to find a particular action.

**Note:** Once you have the action UUID, you can use requests from the actions endpoint to post changes to that action.

## Getting an Unfiltered Array of Actions

Gets a list of current actions in your environment associated to entities discovered by the specified target.

**Example:** GET <https://10.10.10.10/api/v3/targets/xxwWYAE3EeiNKPbzHdETIA/actions>

### Getting a Filtered Array of Actions

Gets a filtered list of actions discovered by a single target. Pass an inputDTO containing one or more of the following criteria:

- actionModeList
  - DISABLED:  
Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.
  - RECOMMENDED:  
Recommend the action so a user can execute it via the given hypervisor or by other external means
  - MANUAL:  
Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
  - AUTOMATED:  
You have directed Turbonomic to execute the action automatically.
  - COLLECTION (**UNUSED**):  
Legacy action mode, no longer used.
- actionStateList
  - DISABLED:  
An action whose policy has been disabled.
  - RECOMMEND:  
An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - PENDING\_ACCEPT:  
When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - ACCEPTED:  
An action that has been accepted, but is not yet in progress.
  - IN\_PROGRESS:  
An action that is being executed.
  - SUCCEEDED:  
An action that was accepted and completed successfully.
  - FAILED:  
An action that was accepted and did not complete successfully.
  - REJECTED:  
When the action mode is set to manual, an action that has been rejected by the user.
  - CLEARED:  
When the action mode is set to manual, an action that is no longer recommended by the market.
  - ACCOUNTING:  
For cloud entities, an action to resize in order to use a different instance template.
  - QUEUED:

When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.

- `actionTypeList`
  - `START`:  
Start an entity, and add it to the Turbonomic market.
  - `MOVE`:  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - `SUSPEND`:  
Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.
  - `ADD_PROVIDER`:  
This is equivalent to the `START` action, but for storage entities.
  - `CHANGE`:  
Move a virtual machine's storage between datastores.
  - `PROVISION`:  
Add capacity to your environment.
  - `RECONFIGURE`:  
Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a `RECONFIGURE` action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.
  - `DELETE`:  
This is equivalent to the `SUSPEND` action, but for storage entities.
  - `RESERVE_ON_PM`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.
  - `RESERVE_ON_DS`:  
When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.
  - `RESIZE_FOR_EFFICIENCY`:  
A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.
  - `RESIZE_FOR_PERFORMANCE`:  
A resize that increases the capacity of a commodity, such as vCPU, or vMEM.
- `cleared`  
Default: `false`.  
When `true`, actions that were removed from the action list before being executed will be returned.
- `costType`
  - `Saving`  
An action that will decrease cloud spending.
  - `Investment`  
An action that will increase cloud spending.

- cumulative

Default: false.

When true, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

- environmentType

- ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

- CLOUD:

Returns only actions related to entities that are part of your cloud environment.

- groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

- hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

**Example:** POST [https://10.10.10.10/api/v3/targets/\\_\\_xwWYAE3EeiNKPbzHdETIA/actions](https://10.10.10.10/api/v3/targets/__xwWYAE3EeiNKPbzHdETIA/actions)

Example inputDto, filtering for actions that are pending acceptance:

```
{
 "actionStateList": [
 "PENDING_ACCEPT"
]
}
```

**Response:** An array of actions applicable to the specified target that are in the PENDING\_ACCEPT state.

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/_uictIJuHEeics6ASgJ-kdg"
 },
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/actions/2204259249360"
 }
],
 "uuid": "_uictIJuHEeics6ASgJ-kdg",
 "createTime": "2018-08-08T23:53:08-04:00",
 "actionType": "RIGHT_SIZE",
 }
]
```

```

"actionState": "PENDING_ACCEPT",
"actionMode": "MANUAL",
"details": "Scale down VMem for VirtualMachine btc-1 from 8 GB to 7 GB",
"importance": 0,
"target": {
 "uuid": "4223dcd6-0b28-6ae6-1066-689912ced3a1",
 "displayName": "btc-1",
 "className": "VirtualMachine",
 "aspects": {
 "virtualMachineAspect": {
 "os": "Other Linux (64-bit)",
 "connectedNetworks": [
 {
 "uuid": "74c91ba4fa390030f47f91e6e1ee77ff7b4db353",
 "displayName": "VM Network"
 }
]
 },
 "numVCPU": 1,
 "ebsOptimized": false
 }
},
"environmentType": "ONPREM"
},
"currentEntity": {
 "uuid": "_ui4yAZuHEeics6ASgJ-kdg",
 "className": "VMem"
},
"newEntity": {
 "uuid": "_ui4yA5uHEeics6ASgJ-kdg",
 "className": "VMem"
},
"currentValue": "8388608.0",
"newValue": "7340032.0",
"resizeToValue": "7340032.0",
"risk": {
 "uuid": "_ujB78ZuHEeics6ASgJ-kdg",
 "subCategory": "Efficiency Improvement",
 "description": "Underutilized VMem in Virtual Machine 'btc-1'",
 "severity": "MINOR",
 "reasonCommodity": "VMem",
 "importance": 0
},
"stats": [
 {
 "name": "costPrice",
 "filters": [
 {
 "type": "savingsType",
 "value": "savings"
 }
]
 },
 {
 "units": "$",
 "value": 50
 }
]
},
"actionID": "2204259249360"
},
...
}
]

```

## Entities For a Specified Target

*Referenced Endpoint:* [Entities](#)

Gets a list of entities discovered by the specified target. This is used most often for reporting, dashboard, or inventory purposes.

### Getting An Array of Entities

Gets a list of all entities discovered by the specified target. An entity is any device in your environment that Turbonomic manages, such as applications, physical machines, VMs, storage, and datacenters.

Example: GET <https://10.10.10.10/api/v3/targets/xxwWYAE3EeiNKPbzHdETIA/actions>

**Response:** An array of EntityApiDTO objects belonging to a vCenter target:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/vsphere-mycorp.com-ResourcePool-resgroup-8?include_aspects=false"
 }
],
 "uuid": "vsphere-mycorp.com-ResourcePool-resgroup-8",
 "displayName": "Resources-Cluster1",
 "className": "VirtualDataCenter",
 "priceIndex": 1e-12,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "__xwWYAE3EeiNKPbzHdETIA",
 "displayName": "vsphere-mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "resgroup-8",
 "environmentType": "ONPREM"
 },
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/entities/vsphere-mycorp.com-GuestLoad-421ddc94-8d0c-504a-62f5-493802e96d0a?include_aspects=false"
 }
],
 "uuid": "vsphere-mycorp.com-GuestLoad-421ddc94-8d0c-504a-62f5-493802e96d0a",
 "displayName": "GuestLoad[Nexus1000V]",
 "className": "Application",
 "priceIndex": 1,
 "state": "ACTIVE",
 "severity": "Normal",
 "discoveredBy": {
 "uuid": "__xwWYAE3EeiNKPbzHdETIA",
 "displayName": "vsphere-mycorp.com",
 "type": "vCenter"
 },
 "remoteId": "vm-51",
 "environmentType": "ONPREM"
 },
 ...
]
```

## Templates Endpoint

Turbonomic uses templates to reserve resources and deploy workload in your environment, to calculate supply or demand changes in a plan, and to calculate workloads for cloud environments.

Using the templates endpoint, you can:

- Get a list of templates available to your Turbonomic appliance
- Get information needed to create templates, such as supported OSes and CPUs
- Create a template
- Edit and Delete templates

## Templates Utility Requests

Utility requests provide metadata that you can use to make related requests, assemble inputDTOs, or see the potential values of a particular DTO.

The templates endpoint contains utility requests that show the possible values for several optional fields used to create detailed templates. These can be used to find the supported cloud OSes and CPU models.



## Getting a List of CPU Models

Gets a list of valid CPU Models. If given a CPU Model, Turbonomic can more accurately reflect your CPU capacity in market decisions.

**Example:** GET <https://10.10.10.10/api/v3/templates/physicalmachine/cpumodels>

**Result:** A list of CpuModelApiDTOs, describing the cores, speed, and internal scaling factor for each CPU. This information will be used to provide more granular CPU data, if passed as part of a TemplateApiInputDTO:

```
[
 {
 "numCores": 4,
 "speed": 3500,
 "scalingFactor": 1.116909571123129,
 "modelName": "AMD A10 PRO-7800B"
 },
 {
 "numCores": 4,
 "speed": 3700,
 "scalingFactor": 0.9745581799084503,
 "modelName": "AMD A10 PRO-7850B"
 },
 ...
 {
 "numCores": 8,
 "speed": 1415,
 "scalingFactor": 1.1638948014750774,
 "modelName": "UltraSPARC T2 Plus"
 }
]
```

## Templates Requests

Template requests handle creation, configuration, editing and deletion of templates on your Turbonomic appliance.

### Getting a List of Templates

Gets a list of templates available to your Turbonomic instance. This list will include both templates discovered from underlying targets and those created through the Turbonomic UI. To get a specific template, pass the template UUID as part of the request.

#### Examples:

- All Templates: GET <https://10.10.10.10/api/v3/templates>
- Single Target: GET [https://10.10.10.10/api/v3/templates/\\_v0Q70MpiEd-hypXfJzX8Wg](https://10.10.10.10/api/v3/templates/_v0Q70MpiEd-hypXfJzX8Wg)

**Response:** A list of TemplateApiDTOs:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/templates/_v0Q70MpiEd-hypXfJzX8Wg"
 }
],
 "uuid": "_v0Q70MpiEd-hypXfJzX8Wg",
 "displayName": "Small",
 "className": "PhysicalMachineProfile",
 "price": 5000,
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCores",
 "value": 4
 },
 {
 "name": "cpuSpeed",
 "units": "MHz",

```

```

 "value": 2992
 },
 {
 "name": "ioThroughputSize",
 "units": "MB/s",
 "value": 890
 },
 {
 "name": "memorySize",
 "units": "MB",
 "value": 32759
 },
 {
 "name": "networkThroughputSize",
 "units": "MB/s",
 "value": 512
 }
]
 },
 "infrastructureResources": [
 {
 "stats": [
 {
 "name": "powerSize",
 "value": 1
 },
 {
 "name": "spaceSize",
 "value": 1
 },
 {
 "name": "coolingSize",
 "value": 1
 }
]
 }
],
 "discovered": false
},
{
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/templates/_v0Q70cpiEd-hypXfJzX8Wg"
 }
],
 "uuid": "_v0Q70cpiEd-hypXfJzX8Wg",
 "displayName": "Medium",
 "className": "PhysicalMachineProfile",
 "price": 10000,
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCores",
 "value": 4
 },
 {
 "name": "cpuSpeed",
 "units": "MHz",
 "value": 2992
 },
 {
 "name": "ioThroughputSize",
 "units": "MB/s",
 "value": 1780
 },
 {
 "name": "memorySize",
 "units": "MB",
 "value": 65519
 },
 {
 "name": "networkThroughputSize",
 "units": "MB/s",
 "value": 512
 }
]
 }
]
},
"infrastructureResources": [

```

```

 {
 "stats": [
 {
 "name": "powerSize",
 "value": 1
 },
 {
 "name": "spaceSize",
 "value": 1
 },
 {
 "name": "coolingSize",
 "value": 1
 }
]
 },
 "discovered": false
 },
 ...
]

```

## Creating a Template

Creates a template for use in your environment. You must pass a completed `TemplateApiInputDTO` as the inputDTO. You can create templates for virtual machines, physical machines, storage, and containers.

**Example:** POST <https://10.10.10.10/api/v3/templates>

### Example inputs:

- Container

```

{
 "computeResources": [
 {
 "stats": [
 {
 "name": "ioThroughput",
 "value": 0
 },
 {
 "name": "networkThroughput",
 "value": 0
 },
 {
 "name": "memorySize",
 "value": 8192
 },
 {
 "name": "cpuSpeed",
 "value": 1536
 }
]
 }
],
 "storageResources": [
 {
 "stats": [
 {
 "name": "diskSize",
 "value": 20
 }
]
 }
],
 "className": "ContainerProfile",
 "displayName": "ContainerTemplate1"
}

```

- Physical Machine

```

{

```

```

"cpuModel": "AMD A10-5700",
"computeResources": [
 {
 "stats": [
 {
 "name": "ioThroughputSize",
 "value": 1000
 },
 {
 "name": "networkThroughputSize",
 "value": 1000
 },
 {
 "name": "memorySize",
 "value": 8192
 },
 {
 "name": "numOfCores",
 "value": 4
 },
 {
 "name": "cpuSpeed",
 "value": "3400"
 }
]
 }
],
"price": 0,
"className": "PhysicalMachineProfile",
"displayname": "HostTemplate1"
}

```

- Storage

```

{
 "storageResources": [
 {
 "stats": [
 {
 "name": "diskIops",
 "value": 250
 },
 {
 "name": "diskSize",
 "value": 750
 }
]
 }
],
 "price": 0,
 "className": "StorageProfile",
 "displayname": "StorageTemplate1"
}

```

- Virtual Machine

```

{
 "computeResources": [
 {
 "stats": [
 {
 "name": "ioThroughput",
 "value": 0
 },
 {
 "name": "networkThroughput",
 "value": 500
 },
 {
 "name": "memorySize",
 "value": 4096
 },
 {
 "name": "memoryConsumedFactor",
 "value": 70
 }
]
 }
],

```

```

 {
 "name": "numOfCpu",
 "value": 2
 },
 {
 "name": "cpuSpeed",
 "value": 1024
 },
 {
 "name": "cpuConsumedFactor",
 "value": 75
 }
]
},
"storageResources": [
 {
 "type": "disk",
 "stats": [
 {
 "name": "diskIops",
 "value": 50
 },
 {
 "name": "diskSize",
 "value": 20
 },
 {
 "name": "diskConsumedFactor",
 "value": 70
 }
]
 }
]
}

```

**Response:** The `TemplateApiInputDTO` for the template created. In this case, a virtual machine template:

```

{
 "uuid": "_lmaQoLUtEei5nORDMStFGA",
 "displayName": "VirtualMachineTemplate1",
 "className": "VirtualMachineProfile",
 "computeResources": [
 {
 "stats": [
 {
 "name": "numOfCpu",
 "value": 2.0
 },
 {
 "name": "cpuSpeed",
 "units": "MHz",
 "value": 1024.0
 },
 {
 "name": "cpuConsumedFactor",
 "units": "%",
 "value": 75.0
 },
 {
 "name": "memorySize",
 "units": "MB",
 "value": 4096.0
 },
 {
 "name": "memoryConsumedFactor",
 "units": "%",
 "value": 70.0
 },
 {
 "name": "ioThroughput",
 "units": "MB/s",
 "value": 0.0
 },
 {
 "name": "networkThroughput",
 "units": "MB/s",
 "value": 500.0
 }
]
 }
]
}

```

```

 }
]
},
"storageResources": [
 {
 "stats": [
 {
 "name": "diskIops",
 "value": 50.0
 },
 {
 "name": "diskSize",
 "units": "GB",
 "value": 20.0
 },
 {
 "name": "diskConsumedFactor",
 "units": "%",
 "value": 70.0
 }
],
 "type": "disk"
 }
],
"deploymentProfile": {
 "uuid": "_1lbZMrUtEei5nORDMStFGA",
 "displayName": "VirtualMachineTemplate1_DeploymentProfile",
 "className": "ServiceCatalogItem",
 "deployParameters": [
 {
 "targetType": "vCenter",
 "providers": [
 {
 "provider": {
 "uuid": "dfe657268310d8b128d441b2a34ec0721e72ee7e",
 "displayName": "Development DC7",
 "className": "DataCenter",
 "discoveredBy": {
 "uuid": "_0jF3UJwFEei_xJ5Ai6-rnQ",
 "displayName": "vsphere-dc7.mycorp.com",
 "type": "vCenter"
 }
 }
 }
],
 "parameters": [
 {
 "parameterType": "image",
 "properties": [
 {
 "name": "id",
 "value": "path/to/image"
 }
]
 }
]
 }
]
},
"discovered": false
}

```

## Editing a Template

Edits a user-created template, given the template UUID and the edited DTO. To edit discovered templates, edit them on the target from which they originate. Note that you must pass a complete TemplateApiDTO with the edited fields.

**Example:** PUT [https://10.10.10.10/api/v3/templates/\\_1maQoLUtEei5nORDMStFGA](https://10.10.10.10/api/v3/templates/_1maQoLUtEei5nORDMStFGA)

**Example input:** Note that both the displayName and diskConsumedFactor values were changed:

```

{
 "computeResources": [
 {
 "stats": [

```

```

 {
 "name": "ioThroughput",
 "value": 0
 },
 {
 "name": "networkThroughput",
 "value": 500
 },
 {
 "name": "memorySize",
 "value": 4096
 },
 {
 "name": "memoryConsumedFactor",
 "value": 70
 },
 {
 "name": "numOfCpu",
 "value": 2
 },
 {
 "name": "cpuSpeed",
 "value": 1024
 },
 {
 "name": "cpuConsumedFactor",
 "value": 75
 }
]
},
"storageResources": [
 {
 "type": "disk",
 "stats": [
 {
 "name": "diskIops",
 "value": 50
 },
 {
 "name": "diskSize",
 "value": 20
 },
 {
 "name": "diskConsumedFactor",
 "value": 75
 }
]
 }
]
},
"className": "VirtualMachineProfile",
"displayname": "VirtualMachineTemplate1 Modified",
"deploymentProfileId": "_1lbZMrUtEei5nORDMStFGA"
}

```

**Response:** The modified TemplateApiDTO reflecting the submitted changes.

## Deleting a Template

Deletes the specified template. This will not delete associated deployment profiles, and deployments already scheduled to use this template will not succeed.

**Example:** DELETE [https://10.10.10.10/api/v3/templates/\\_1maQoLUtEei5nORDMStFGA](https://10.10.10.10/api/v3/templates/_1maQoLUtEei5nORDMStFGA)

**Response:** A successful deletion will return true.

## Topology Definitions Endpoint

Turbonomic gives you the ability to create your own Business Applications, Business Transactions, and Services without the need to ingest additional application data into the Turbonomic platform. This is especially useful in cases where there are gaps in the application stack shown in the Turbonomic supply chain. For example, in the absence of an application monitoring target such as AppDynamics or Dynatrace, you will not see Business Applications in your supply chain. This feature addresses those gaps.

When you create a new application entity, you identify interrelated application entities and nodes in your existing environment for which you want to measure performance, so Turbonomic can link them in a supply chain and represent them as a unified group. You can monitor overall performance for the group in the context of the new application entity, and drill down to the individual entities and nodes for finer details.

Using the topology definitions endpoint of the Turbonomic API, you can:

- Get a single topology definition or list of topology definitions present in your environment
- Create a topology definition
- Edit a topology definition
- Delete a topology definition

## Topology Definition Requests

Topology definition requests handle creation, configuration, editing, and deletion of topology definitions on your Turbonomic appliance.

### Getting Topology Definitions

For a list of topology definitions, the API returns an array of `TopologyDataDefinitionApiDTO`. Each `TopologyDataDefinitionApiDTO` will contain the UUID, entity type, and connection attributes for the defined topology. To get a single topology definition, include the UUID of the topology definition in the request.

#### Examples:

- All Topology Definitions: GET <https://10.10.10.10/api/v3/topologydefinitions>
- Single Topology Definition: GET <https://10.10.10.10/api/v3/topologydefinitions/284711321829968>

#### Response:

```
{
 "uuid": "284711313871856",
 "displayName": "Daltest1__App",
 "entityType": "BusinessApplication",
 "entityDefinitionData": {
 "namePrefix": "Daltest1_",
 "entityType": "VirtualMachine",
 "generationAndConnectionAttribute": {
 "expVal": "App",
 "expType": "EQ",
 "filterType": "virtualMachinesByTag",
 "caseSensitive": true,
 "entityType": null,
 "singleLine": false
 }
 }
}
```

## Creating a Manual Topology Definition

To create a manual topology definition, you must provide the `entityType` of the definition, and manual mapping data to the entities to use. Consider the following input DTO to create a manual topology definition:

```
{
 "entityType": "BusinessApplication",
 "entityDefinitionData": {
 "manualConnectionData": {
 "BusinessTransaction": {
 "staticConnections": [
 "73585155548499",
 "73585155548500"
]
 }
 }
 }
}
```



```

 "Service":{
 "staticConnections":[
 "73585155548487",
 "73585155608929"
]
 },
 "ApplicationComponent":{
 "staticConnections":[
 "73596151905427",
 "73596151905425",
 "73596151905428",
 "73596151905429"
]
 },
 "VirtualMachine":{
 "staticConnections":[
 "73586657216608",
 "73585300874097",
 "73586657216576"
]
 },
 "DatabaseServer":{
 "staticConnections":[
 "73585314633044",
 "73585314632869",
 "73585314632946"
]
 },
 "Container":{
 "staticConnections":[
 "73585323455544",
 "73582124234229",
 "73585385438946"
]
 },
 "ContainerPod":{
 "staticConnections":[
 "73585334243044",
 "73585314324535",
 "73585314533256"
]
 }
 },
 "displayName":"DalManualtest_"
}

```

In the manualConnectionData parameter of the entityDefinitionData, the UUIDs of the related entities have been provided for each level of the supply chain for which the topology should be instantiated.

The creation can be as simple or complex as desired. For example, a simple manual topology may look like the following:

```

{
 "entityType":"BusinessTransaction",
 "entityDefinitionData":{
 "manualConnectionData":{
 "Service":{
 "staticConnections":[
 "73585155548487"
]
 },
 "Container":{
 "staticConnections":[
 "73596151905298"
]
 }
 }
 },
 "uuid":"284711514573216",
 "displayName":"DalManualSimple"
}

```

## Creating an Automated Topology Definition

To create an automated topology definition, you must provide the `entityType` of the definition, and the data that makes up the regular expression used to determine dynamic group membership. Consider the following input DTO to create an automated topology definition:

```
{
 "entityType": "BusinessApplication",
 "entityDefinitionData": {
 "entityType": "VirtualMachine",
 "namePrefix": "Daltest1_",
 "generationAndConnectionAttribute": {
 "expType": "EQ",
 "expVal": "App",
 "filterType": "vmsByTag",
 "caseSensitive": false
 }
 }
}
```

In the `generationAndConnectionAttribute` parameter of the `entityDefinitionData`, the information that makes up the regular expression is sent. In this instance, it will match any virtual machine whose tag value equals "App".

## Editing a Topology Definition

Edits the input fields of an existing topology definition, and accepts the modified `TopologyDataDefinitionApiDTO` as input.

**Note:** You can GET the topology definition first using the GET `/topologydefinition/UUID` request to see the current configuration. Note that you must pass all fields in the modified `TopologyDataDefinitionApiDTO`, not only the changed fields.

A successful edit will return the modified `TopologyDataDefinitionApiDTO`.

**Example:** PUT `https://10.10.10.10/api/v3/topologydefinitions`

## Deleting a Topology Definition

Deletes the specified topology definition and removes all created entities. This will also remove any pending actions related to the deleted entities.

**Example:** DELETE `https://10.10.10.10/api/v3/topologydefinitions/284711321829968`

**Response:** A successful deletion request returns a Response Code of 200.

## Workflows Endpoint

When added to your Turbonomic installation, Orchestrator targets assign workflows that execute multiple actions to make changes in your environment. Turbonomic discovers workflows that you have defined on the orchestrator. You can then set up an automation policy that maps workflows to actions. If the action mode is Manual or Automatic, then when Turbonomic recommends the action, it will direct the orchestrator to use the mapped workflow to execute it.

For more information about specific Orchestrator targets, see [Orchestrator Targets](#). For more information about execution of workflows, see [Action Orchestration](#).

Using the workflows endpoint, you can:

- Get a list of all workflows
- Get detailed information about a specified workflow

## Workflows Requests

### Getting Workflows

For a list of workflows, the API returns an array of WorkflowApiDTOs. Each WorkflowApiDTO object will contain detailed information about that workflow, including description and parameters. To get a single workflow, include the UUID of the workflow in the request.

This call takes the following parameter:

type

Filters the request by type of workflow. [ UCSD, AWS\_Service\_Catalog ]

#### Examples:

- All Workflows: GET <https://10.10.10.10/api/v3/workflows>
- Single Workflow: GET <https://10.10.10.10/api/v3/workflows/UCSD-WF-ucsd.eng.mycorp.com-103>

#### Response:

```
[
 {
 "links": [
 {
 "rel": "self",
 "href": "https://10.10.10.10/api/v3/workflows/UCSD-WF-ucsd.eng.mycorp.com-103"
 }
],
 "uuid": "UCSD-WF-ucsd.eng.mycorp.com-103",
 "displayName": "APIC Container Delete Firewall Rule Action",
 "className": "Workflow",
 "actionType": "NONE",
 "parameters": [
 {
 "name": "APIC_TENANT_L4L7_EPG_ACE_IDENTITY",
 "isMandatory": true,
 "isSecret": false,
 "specificValueType": "ApicTenantL4L7EPGAceIdentity"
 },
 {
 "name": "L4L7Identity",
 "isMandatory": true,
 "isSecret": false,
 "specificValueType": "gen_text_input"
 },
 {
 "name": "ACLName",
 "isMandatory": true,
 "isSecret": false,
 "specificValueType": "gen_text_input"
 },
 {
 "name": "ACENAME",
 "isMandatory": true,
 "isSecret": false,
 "specificValueType": "gen_text_input"
 }
],
 "discoveredBy": {
 "uuid": "_U1MIJv-Eei_xJ5Ai6-rnQ",
 "displayName": "ucsd.eng.mycorp.com",
 "type": "Cisco UCS Director"
 }
 },
 ...
]
```

## Getting Workflows of a Specific Type

By adding the type parameter to the `/api/v3/workflows` request, you can filter workflows from a specific source. For example, the response to this request will be an array of `WorkflowApiDTOs` discovered through UCS Director targets:

**Example:** GET `https://10.10.10.10/api/v3/workflows?type=UCSD`

## Turbonomic Cookbook

---

### Purpose

The Turbonomic cookbook is intended to assist in performing specific tasks using the API, and answering commonly-asked questions while presenting Turbonomic API best practices. This section of the API Guide is a living document, meaning that it is constantly being added to.

If there is a situation or task you would like to see included, please log your request via the IBM Ideas portal located at:

<https://ideas.ibm.com/>

### Cookbooks

Each cookbook contains one or more recipes. Each recipe will contain all the necessary information to complete the specified task, along with a basic script showing the functionality of the recipe.

- [Authenticating with the Turbonomic API](#)
- [Plan Results Cookbook](#)
  - [Seeing Changes to a Subset of Entities](#)

## Authenticating With the API

### Objective

To obtain an authentication token after logging in with the API, and use that token on subsequent API calls.

### Authentication in the API

To use the API, you must have a valid user account on the Turbonomic instance. Also note that accounts can have different roles. The API will only execute commands that are valid for your user role. For example, to execute Turbonomic recommended actions, your account must have a role of either administrator, deployer, or automator.

To make API calls, you request an authentication token and pass it with each call to the Turbonomic API. The token request returns a cookie for your authentication. A common way to use this token is to store the cookie locally, and pass it with your API calls.

**Example:** `curl -s -k -c /tmp/cookies -H 'accept: application/json' 'https://localhost/api/v3/login?hateoas=true' -d 'username=administrator&password=password'`

Then, each request must use the `-b cookie-filename` parameter to use the session cookie delivered by the login request.

Another approach is to get the authentication header and parse out the authentication cookie. Then you can create a header for each API request that includes the cookie. For example, assume you store the value in a variable named token. You could use it like this:

```
headers = {'cookie': token}
r = requests.get('https://10.10.123.456/api/v3/targets/specs', headers=headers, verify=False,
stream=True)
```

## Prerequisites

In order to obtain an authentication token, you must have the following information:

- The IP address of the Turbonomic instance.
- The username and password of the user who will be logged in.

## Procedure

To obtain an authentication token:

1. Construct the API payload.

The input body should be JSON, in the following format with your credentials in place of myUsername and myPassword:

```
{'username': 'myUsername', 'password': 'myPassword'}
```

2. Make the API request.

Use the POST `https://[INSTANCE_IP]/api/v3/login` request, with the input body constructed in the previous step.

3. Retrieve the authentication token from the response headers.

Example headers:

```
{
 'Server': 'nginx',
 'Date': 'Mon, 01 Feb 2021 20:17:02 GMT',
 'Content-Type': 'application/json',
 'Transfer-Encoding': 'chunked',
 'Connection': 'keep-alive',
 'Vary': 'Accept-Encoding',
 'Set-Cookie': 'JSESSIONID=node0jzerbqte09pe1j29x6ypo92fh17.node0; Path=/; HTTPOnly; Secure',
 'Expires': 'Thu, 01 Jan 1970 00:00:00 GMT',
 'X-Content-Type-Options': 'nosniff',
 'X-XSS-Protection': '1; mode=block',
 'Strict-Transport-Security': 'max-age=63072000; includeSubDomains',
 'X-Frame-Options': 'SAMEORIGIN',
 'X-Turbo-Upstream': 'API',
 'Content-Encoding': 'gzip'
}
```

The authentication token is the first ;-separated token in the value of the Set-Cookie header. In this example, it is `JSESSIONID=node0jzerbqte09pe1j29x6ypo92fh17.node0`.

## Using the Authentication Token in Subsequent API Requests

Once you have obtained an authentication token, you can use that token to make further API requests. To do so, include a header in your request with the name of cookie and a value of the token. For example:

```
{'cookie': 'JSESSIONID=node0jzerbqte09pe1j29x6ypo92fh17.node0' }
```

## Script Example

These listings show a function that gets the authentication token from the cookie, and a script that invokes the function and prints out the token value.

### Function to get the token:

To invoke this function, pass the IP address, the username, and password as arguments. The function builds the authentication payload and requests the token from the given Turbonomic instance. It then parses out the first token in the Set-Cookie header and returns that value.

```
import json
import requests
import urllib3
urllib3.disable_warnings()

def get_cookie(ip, username, password):
 payload = {'username': username, 'password': password}
 r = requests.post(f'https://{ip}/api/v3/login', data=payload, verify=False)
 r.encoding = 'JSON'
 rh = r.headers
 token = rh['Set-Cookie'].split(';')[0]

 return token
```

### Script that calls the function:

Assume the script is saved as `use_login.py`. Also assume the function is in the file `./login.py`. To invoke this script, open a shell at the script file location and enter:

```
./use_login.py <My_IP_Address> <My_Username> <My_Pwd>
```

where the three arguments are the IP address of your Turbonomic, your username, and your password, respectively. The script imports the `./login.py` file and checks for four arguments (the script name and the three arguments you passed). It then executes the `login.get_cookie()` function. The function returns a token, which the script then prints out.

```
#!/usr/bin/env python3

import sys
import login

if len(sys.argv) != 4:
 print("Incorrect arguments list.")
 sys.exit()

token = login.get_cookie(sys.argv[1], sys.argv[2], sys.argv[3])
print(f'TOKEN IS: {token}')
```

## Script Result

The result of this script displays the authentication token:

```
TOKEN IS: JSESSIONID=node0jzzerbqte09pe1j29x6ypo92fh17.node0
```

## Plan Results Cookbook

This cookbook explores plan markets in Turbonomic. Interpreting and parsing this plan market will enable you to better understand how and why Turbonomic has performed a certain action on a particular entity, how the resources- both used and available- have changed, and more as recipes are added to the cookbook.

**Note:** The Turbonomic UI has certain plan configuration options, available to present a base set of configuration changes to accomplish a particular goal. For example, the Add Workload plan is preconfigured to expect additional workloads to be provisioned. To see how Turbonomic creates these configurations, view the DTOs for the UI plan configurations at [Scenario Examples](#).

## Related Endpoints

When interpreting plan results, the following endpoints are commonly used to retrieve and clarify data:

- **Actions:** Retrieving the actions for the plan market will enable you to see what entities were directly affected during the run of the plan.
- **Entities:** Retrieving the entities for the plan market will enable you to see a full listing of the entities involved in the plan.
- **Groups:** Retrieving the groups for the plan market will enable you to more easily obtain UUID information for the entities in the specified group.
- **Statistics:** Retrieving the market statistics for the plan market as a whole, a group, or an entity will enable you to easily see a broad overview of the state of the entity utilization after the plan has finished.

## Use Case Index

The Plan Results Cookbook has the following recipes:

- [Seeing Changes to a Subset of Entities](#)

## Seeing Changes to a Subset of Entities

### Objective

After running a plan, you may want to see the effects of your plan on a particular entity or group of entities. This can help you better understand the effect your configuration changes would have on a subset of plan entities, or to explore why Turbonomic made specific action recommendations.

For example, assume that there is a virtual machine that shares dependencies with a number of other entities. Apart from the overall plan results, you might want to take a closer look at how the plan impacts this VM in particular.

### Prerequisites

In order to investigate the changes to the specified entity, you need to know:

- The UUID of a successfully completed plan market

To obtain this UUID, either get it from the API response sent when you POST a scenario to the market, or execute the GET `/markets` request, and find the market whose name corresponds with the name of the scenario. Alternately, you can find the scenario's UUID in the `scenario` parameter of the plan market. Verify that the `state` parameter has a value of `SUCCEEDED`. For example, the UUID of the scenario that created the following plan market is `214410933912976`, and the UUID of the plan market itself is `214410933923968`:

```
{
 "uuid": "214410933923968",
 "displayName": "Optimize Cloud 2",
 "className": "Market",
 "state": "SUCCEEDED",
 "stateProgress": 100,
 "scenario": {
 "uuid": "214410933912976",
 "displayName": "Optimize Cloud 2",
 ...
 }
 ...
}
```

For cases where the same scenario has been applied to multiple plans, be sure to check the plan's `runDate` parameter corresponds to the time you expect the plan to have run.

For more information about creating and running scenarios, see [Scenario DTO Breakdown](#).

- The UUID(s) of the entities you want to inspect

Entity UUIDs in a plan market are identical to their real-time market counterparts. This means you can get the UUIDs either from searching the plan market for a related field (for example, the `displayName` of the entity) or searching in the Turbonomic user interface to find the entity UUID.

You can also use the API's Search endpoint to search for entities by different criteria.

## Related Endpoints

When interpreting plan results, the following endpoints are commonly used to retrieve and clarify data:

- **Actions:** Retrieving the actions for the plan market will enable you to see what entities were directly affected during the run of the plan.
- **Entities:** Retrieving the entities for the plan market will enable you to see a full listing of the entities involved in the plan.
- **Groups:** Retrieving the groups for the plan market will enable you to more easily obtain UUID information for the entities in the specified group.
- **Statistics:** Retrieving the market statistics for the plan market as a whole, a group, or an entity will enable you to easily see a broad overview of the state of the entity utilization after the plan has finished.

## Procedure

To investigate changes to a particular subset of plan entities:

1. Get the list of UUIDs you want investigate.

To get all the entities in a market:

```
markets/{market_uuid}/entities
```

If the scope of the plan is small, you can use this to get all the entities in the plan, and then iterate through the results for other attributes such as entity type or display name.

```
markets/{market_uuid}/entities
```

For a large scope, you can use the Search endpoint to find VMs by specific criteria. This works because the plan market and the real-time market use the same UUID values for the same entities. For example, the following call gets all VMs that have the string F00 in their names:

```
https://10.10.123.456/api/v3/search?q=F00&types=VirtualMachine&entity_types=VirtualMachine&query_type=CONTAINS
```

2. Get the list of plan actions.

Run GET `/markets/{market_Uuid}/actions` against the plan market. You can also use POST `/markets/{market_uuid}/actions` to post a filter for advanced action queries, such as only returning actions of a particular type, or particular risk severity.

Another way to get plan actions is to use the Search endpoint. You can POST search parameters that limit the returned actions by text in the description, cost type, or other data. For example, the following search returns only actions on VMs:

```
searchSpec = {
 "actionRelationTypeFilter": [
 "NONE"
],
 "costType": "SAVING",
 "descriptionQuery": {
 "query": "Virtual Machine",
 "type": "CONTAINS",
 "caseSensitive": "true"
 }
}
```



```
 "detailLevel": "STANDARD"
}
```

With the above payload, the following POST would return VM actions from the plan:

```
requests.post(f'https://{ip}/api/v3/markets/{marketUuid}/actions',
 headers=headers,
 json = searchSpec,
 verify=False,
 stream=True)
```

3. Search the API response and find the action(s) relevant to your entities.

Each action will have a `target` parameter, which will in turn have a `uuid`. This UUID identifies the entity the action executed on. This is the best way to locate your entity in a list of actions.

If no actions have the specified `target > uuid` value, then no actions were performed on the entity in question.

**Note:** You can use `GET /markets/{market_Uuid}/entities` to verify that the UUID you are searching for exists in the plan market. If it does not, it means that the entity was not included in the scope of the plan, or the entity is considered an idle workload. An idle workload shows no utilization, so the plan cannot determine optimal placement or what percentage of allocated resources that workload will require when it restarts.

4. Analyze the results.

Each action contains the information detailed in the `ActionApiDTO` data, including the details of the action, old/new values for any changed resources, a record of deletion for DELETE actions, etc.

Search through this DTO and make note of any parameters important to you. (For a DTO reference, see [ActionApiDTO](#).)

## Script Example

The following script searches through the list of actions for a virtual machine, and returns the requested fields for quick comparison. In this case, the UUID, type, state, reason, risk category, and risk subcategory are returned for the action, and paired with the display name, template UUID, and location of the entity.

This script is easily extensible- for example, if `currentLocation > displayName` is equal to `newLocation > displayName`, you may choose not to show either of those fields, as no change involving the VM location was made. You might also use these results to make further calls- for example, using the template UUID to make a `GET /templates/{template_Uuid}` request to return more information about the template being used for a resize action.

Note that the script uses dot-notation to name the fields you want to inspect. It includes the `getFieldFromDict` function to convert those fields to nested Python dictionary values. Because the API returns JSON data, a function like this could be a generally useful utility.

```
#!/usr/bin/env python3

#
Commandline args: platformIP, username, pwd, marketID, VmIDs as comma-sep string (no spaces)
EXAMPLE: ./entity_changes.py 10.10.123.456 MyUser MyPwd 215426452602368
74635124762083,74635124762088,74635124762090
#
import json
import requests
import urllib3
import re

import sys
sys.path.append('../login') #This makes login visible...
import login

#
This function takes json-like dot-notation fields and returns python dictionary values.
That makes it easier to specify the DTO fields you want to view.
#
```

```

def getFieldFromDict(field, dict):
 fToks = field.split('.')
 l = len(fToks)
 workingDict = dict
 for item in fToks:
 if item in workingDict:
 ret = workingDict[item]
 else:
 return "ITEM NOT IN OBJECT"
 l -= 1
 if l == 0:
 return ret
 else:
 workingDict = ret
 return ret

#
This list of fields you want to query, in dot-notation
#
RetrieveFields = [
 "uuid",
 "actionType",
 "actionState",
 "details",
 "currentEntity.displayName",
 "newEntity.displayName",
 "template.uuid",
 "risk.subCategory",
 "risk.description",
 "currentLocation.displayName",
 "newLocation.displayName"
]

#
A search spec to get actions that are on VMs.
#
searchSpec = {
 "actionRelationTypeFilter": [
 "NONE"
],
 "costType": "SAVING",
 "descriptionQuery": {
 "query": "Virtual Machine",
 "type": "CONTAINS",
 "caseSensitive": "true"
 },
 "detailLevel": "STANDARD"
}

#
Make sure you have the correct count of arguments.
#
if len(sys.argv) != 6:
 print("Incorrect arguments list.")
 sys.exit()

#
Get the args into well named vars, and also authenticate.
#
ip = sys.argv[1]
marketUuid = sys.argv[4]
ElemsToInspect = sys.argv[5].split(',')

token = login.get_cookie(ip, sys.argv[2], sys.argv[3])
headers = {'cookie': token}

#
Search plan for actions on VMs
#
actions_r = requests.post(f'https://{ip}/api/v3/markets/{marketUuid}/actions', headers=headers,
 json = searchSpec, verify=False, stream=True)

if actions_r.status_code != 200:
 print(f'Market Request Status is {actions_r.status_code}.')
 sys.exit()

#
For each action that targets one of the entities you pass in the command line,
parse the object and display the field values.
#
print("Plan Changes for Entity(ies):")
print(sys.argv[5])

```

```

print("-----")
r_json = actions_r.json()
for item in r_json:
 target = item['target']
 if 0 < ElementsToInspect.count(target["uuid"]):
 print(f'VM ACTION: {target["displayName"]}')
 for field in RetrieveFields:
 val = getFieldFromDict(field, item)
 print(f' {field}: {val}')

```

## Script Result

The result of this script displays the requested information for each plan entity:

```

Plan Changes for Entity(ies):
74635124762083,74635124762088

VM ACTION: test-rosa-28tfg-worker-us-west-2a-5hr8j
 uuid: 145057709553360
 actionType: SCALE
 actionState: SUCCEEDED
 details: Scale Virtual Machine test-rosa-28tfg-worker-us-west-2a-5hr8j from m5.xlarge to
r5a.large in Advanced Engineering
 currentEntity.displayName: m5.xlarge
 newEntity.displayName: r5a.large
 template.uuid: 74635137532084
 risk.subCategory: Efficiency Improvement
 risk.description: Underutilized VCPU, Net Throughput
 currentLocation.displayName: aws-US West (Oregon)
 newLocation.displayName: aws-US West (Oregon)
VM ACTION: test-rosa-28tfg-infra-us-west-2a-mw867
 uuid: 145057709553376
 actionType: SCALE
 actionState: SUCCEEDED
 details: Scale Virtual Machine test-rosa-28tfg-infra-us-west-2a-mw867 from r5.xlarge to
r5a.xlarge in Advanced Engineering
 currentEntity.displayName: r5.xlarge
 newEntity.displayName: r5a.xlarge
 template.uuid: 74635137531741
 risk.subCategory: Efficiency Improvement
 risk.description: Increase RI Utilization
 currentLocation.displayName: aws-US West (Oregon)
 newLocation.displayName: aws-US West (Oregon)

```

## Seeing the Changes for a Group of Entities

You can easily modify this script to see the changes for a particular group of entities without needing to know each UUID.

To perform this modification, consume the result of a GET `/groups/{group_uuid}/entities` request. Example result:

```

[
 {
 "uuid": "73422419372894",
 "displayName": "PT_RHEL",
 "className": "VirtualMachine",
 "priceIndex": 4.33,
 "state": "ACTIVE",
 "severity": "Normal",
 "costPrice": 0.24104111,
 "discoveredBy": {
 "uuid": "73422416392368",
 "displayName": "producttrust",
 "category": "Public Cloud",
 "type": "AWS",
 "readonly": false
 },
 },
 ...

```

```

 },
 {
 "uuid": "73422419372988",
 "displayName": "PT_Resize_Up_Target",
 "className": "VirtualMachine",
 "priceIndex": 10074.822,
 "state": "ACTIVE",
 "severity": "Normal",
 "costPrice": 0.078387,
 "discoveredBy": {
 "uuid": "73422416392368",
 "displayName": "producttrust",
 "category": "Public Cloud",
 "type": "AWS",
 "readonly": false
 },
 },
 ...
 },
 {
 "uuid": "73422419372921",
 "displayName": "PT_Consistent_Scaling_VM_1",
 "className": "VirtualMachine",
 "priceIndex": 100.357,
 "state": "ACTIVE",
 "severity": "Normal",
 "costPrice": 0.0159137,
 "discoveredBy": {
 "uuid": "73422416392368",
 "displayName": "producttrust",
 "category": "Public Cloud",
 "type": "AWS",
 "readonly": false
 },
 },
 ...
}
]

```

Using this result, you can either use each UUID found, or filter based on entities. For example, to iterate through a group and get the UUID of all VM entities:

```

foreach (var entity in groupApiResponse)
{
 if (entity["className"] == VirtualMachine)
 {
 Console.WriteLine(entity["uuid"])
 }
}

```

## Creating a Static Group

### Objective

To create a static group of entities.

A static group contains a fixed set of entities. This example creates a group of VMs, but you can create a group of any entity type.

The script will get the unique IDs of member entities by their display names, and add them to a list that it then uses to display the group members.

The script also sets up the group type and entity type when it creates the payload data. It then uses the API to create the group, and verify that the group contains the entities you specified.

### Prerequisites

In order to create a static group, you must have the following information:

- The display names of the members in the group
- The entity type you will use for this group

- The IP address and login credentials for a Turbonomic instance

## Related Endpoints

This recipe uses the following endpoints:

- [Groups](#)

It uses the `POST /groups` request to create the group, and the `GET /groups/{group_uuid}` request to verify the group membership.

- [Search](#)

It uses the `GET /search` request to obtain the UUIDs of the group members from the display names.

## Script Setup

To set up the script, provide values from your environment for the following:

- `ip`

The IP address of your Turbonomic instance.

- `username`

The username for the account Turbonomic will use to access the API.

- `password`

The password for the account Turbonomic will use to access the API.

- `names_list`

A list of display names for the entities you want to add to the group. Note that the entities must all be of the same type. If entities of more than one type are provided, group creation will fail.

- `group_name`

The display name for the group. This name will appear in the user interface.

- `entity_type`

The entity type for the group and its member entities. The script will use this in creating the Group input data, and also when executing a search for the entities' UUID.

**Note:**

Turbonomic does not support creation of groups with more than one entity type at this time.

## Authentication

The first step of any script is to authenticate access to your Turbonomic instance, retrieve the authentication token and store it for use with subsequent calls. To authenticate, this script uses the values you set for `ip`, `username`, and `password`. For more detail, see the [Authentication Recipe](#).

**API Request:** `https://{ip}/vmturbo/rest/login`

## UUID Retrieval

After it logs in, the script iterates through your list of entity names, passing each in turn to the `GET /search` request. It uses the returned `EntityApiDTO` to get the UUIDs for each matching entity.

The search response is an array of entity objects, and each entity object includes the entity's UUID. The script loops through the display names you provide in `names_list`, and executes a separate search request for each name. Each response should be an array of one item. The script adds the `uuid` of that item to the `group_members_uuids_list`.

**Note:**

Another method to obtain the UUIDs is to execute a filtered POST `/search` request. This script uses repeated GET `/search` requests for simplicity.

**Sample Request:** GET `https://10.10.10.10/api/v3/search?q=zrlinuxclone106&types=VirtualMachine&entityTypes=VirtualMachine`

**Sample Return:**

```
[
 {
 "uuid": "74011863734144",
 "displayName": "zrlinuxclone106",
 "className": "VirtualMachine",
 "state": "ACTIVE",
 "severity": "Critical",
 "discoveredBy": {
 "uuid": "73935378522272",
 "displayName": "vsphere-dc23.eng.vmturbo.com",
 "category": "Hypervisor",
 "type": "vCenter",
 "readonly": false
 },
 "severityBreakdown": {
 "CRITICAL": 1
 },
 "providers": [
 {
 "uuid": "73935378875792",
 "displayName": "hp-esx108.eng.vmturbo.com",
 "className": "PhysicalMachine"
 },
 {
 "uuid": "73935378875736",
 "displayName": "QS2_GRID01:DC23DS1",
 "className": "Storage"
 },
 {
 "uuid": "73995333587472",
 "displayName": "Resources-hp-esx108.eng.vmturbo.com",
 "className": "VirtualDataCenter"
 }
],
 "environmentType": "ONPREM",
 "vendorIds": {
 "vsphere-dc23.eng.vmturbo.com": "vm-2844"
 }
 }
]
```

## Constructing the Input DTO and Creating the Group

Once the script has built the list of UUIDs for the group members, it can build a `GroupApiDTO` to describe the group you want. The DTO needs values for the following properties:

- `isStatic`: true to create a static group
- `displayName`: The group name that you set for the `group_name` variable
- `memberUuidList`: The array of UUID strings that the script has created
- `criteriaList`: For a static group, pass an empty array
- `groupType`: The entity type for this group -- the value you set for the `entity_type` variable

Using the obtained UUIDs and the above information, the script will construct the input `GroupApiDTO`. The script creates a Python dictionary, which it passes as the input body of the POST `/groups` request.

**Example Input DTO:**

```
{
 "isStatic": "true",
 "displayName": "TestGrp2",
```

```

"memberUuidList":[
 "74011863734144",
 "73997881750336"
],
"criteriaList":[
],
"groupType":"VirtualMachine"
}

```

**Sample Request:** POST <https://{ip}/api/v3/groups/>

The response to this request will be the GroupApiDTO for the created group.

**Sample Response:**

```

{
 "uuid":"285123335341792",
 "displayName":"TestGrp2",
 "className":"Group",
 "membersCount":2,
 "entitiesCount":2,
 "groupType":"VirtualMachine",
 "severity":"CRITICAL",
 "environmentType":"ONPREM",
 "isStatic":true,
 "logicalOperator":"AND",
 "memberUuidList":[
 "74011863734144",
 "73997881750336"
],
 "temporary":false,
 "activeEntitiesCount":1,
 "cloudType":"UNKNOWN",
 "memberTypes":[
 "VirtualMachine"
],
 "entityTypes":[
 "VirtualMachine"
],
 "groupClassName":"GroupApiDTO"
}

```

**Verification**

To verify that the group was created successfully, the script uses the returned UUID to get the group data via the /groups endpoint. If the request gets the group, the returned DTO should be identical to the response from the request to create the group.

If the group does not exist for the given UUID, the API returns the request status 404, with the message, Group not found: <Passed\_UUID>. This indicates that the script failed to create the group.

**Sample Response:** [https://{ip}/api/v3/groups/{group\\_uuid}](https://{ip}/api/v3/groups/{group_uuid})

You can choose to only use a subset of the provided response. For example, this script prints the displayName and UUID for each group member.

**Sample Script Response:**

```

Group Members:
Display name: boao.min-spare
OID: 73997881750336
Display name: zrlinuxclone106
OID: 74011863734144

```

## Script Example

```
import json
import requests
import urllib3
import re

import sys
sys.path.append('../login') #This makes login visible...
import login

#
Make sure you have the correct count of arguments.
#
if len(sys.argv) != 7:
 print("Incorrect arguments list.")
 sys.exit()

#
Get the args into well named vars, and also authenticate.
#
ip = sys.argv[1]
groupName = sys.argv[4]
entityType = sys.argv[5]
entsListByName = sys.argv[6].split(',')

token = login.get_cookie(ip, sys.argv[2], sys.argv[3])
authHeader = {'cookie': token}

#
Get UUIDs for entities in entsListByName
#
memberIds = []
for name in entsListByName:
 response = requests.get(
 f'https://{ip}/api/v3/search?
q={name}&types={entityType}&entity_types={entityType}&query_type=EXACT',
 headers=authHeader, verify=False)
 if response.status_code == 200:
 response_json = json.loads(response.text)
 memberIds.append(response_json[0]['uuid'])
 if response.status_code != 200:
 print(f'Search Request Status for {name} is {response.status_code}.')

#
Assemble the payload object to create the group.
#
group_input_dto = {}
group_input_dto['isStatic'] = 'true'
group_input_dto['displayName'] = groupName
group_input_dto['memberUuidList'] = memberIds
group_input_dto['criteriaList'] = []
group_input_dto['groupType'] = entityType

#
Create the group
#
response = requests.post(f'https://{ip}/api/v3/groups', json=group_input_dto,
headers=authHeader, verify=False, stream=True)

if response.status_code != 200:
 print(f'Create Group failed. StatusCode: {response.status_code}\n{response.text}')
 sys.exit()

groupResponse_json = json.loads(response.text)
groupId = groupResponse_json['uuid']

#
Inspect the group
#
groupCheck_response = requests.get(f'https://{ip}/api/v3/groups/{groupId}', headers=authHeader,
verify=False)
groupCheck_json = json.loads(groupCheck_response.text)

if groupCheck_json['uuid']:
 verifiedId = groupCheck_json['uuid']
 groupName = groupCheck_json['displayName']
 print(f'Group \"{groupName}\" successfully created with a UUID of: {verifiedId}')
else:
```



```

 print('Error in group creation, no UUID found.')
 sys.exit()

Get a list of Group members
response = requests.get(
 f'https://{ip}/api/v3/groups/{groupId}/entities', headers=authHeader, verify=False)
if response.status_code != 200:
 print(f'Could not get group members. Status is {response.status_code}.')
 sys.exit()

groupMembers_json = json.loads(response.text)

Print a list of group members
print('\nGroup Members:')
for item in groupMembers_json:
 print('Display name: ' + item['displayName'])
 print('OID: ' + item['uuid'])

```

## Creating a Dynamic Group

### Objective

To create a dynamic group of entities.

A dynamic group contains a set of entities discovered by filter criteria. In this case, the script will create a group of all virtual machines which reside on a specific host. As virtual machines are added and removed from the host, the membership of this group will reflect those changes.

The script will use the /search endpoint to get the unique ID of the host using its display name, and then retrieve a list of expected virtual machines to verify against. The script then creates the criteria list and inserts that value into the rest of the payload data before creating the group and verifying that the membership is as expected.

### Prerequisites

In order to create a dynamic group, you must have the following information:

- The display name of the host
  - The entity type you will use for this group
  - The IP address and login credentials for a Turbonomic instance

### Related Endpoints

This recipe uses information from the following endpoints:

- [Groups](#)

#### Note:

All possible criterion values can be found in the [Groups Endpoint Tips](#).

- [Search](#)

### Script Setup

To set up the script, provide values from your environment for the following:

- ip
  - The IP of your Turbonomic instance.
- username
  - The username for the account Turbonomic will use to access the API.
- password

The password for the account Turbonomic will use to access the API.

- `host_name`

The display name of the host whose virtual machines will be added to the group.

- `group_name`

The display name for the group. This name will appear in the user interface.

- `entity_type`

The entity type for the member entities of the group. The script will use this in creating the Group input data, and also when verifying group membership.

**Note:**

Turbonomic does not support creation of groups with more than one entity type at this time.

## Authentication

The first step of any script is to authenticate to your Turbonomic instance, retrieve the authentication token and store it for use with subsequent calls. To authenticate, this script uses the values you set for `ip`, `username`, and `password`. For more detail, see the [Authentication Recipe](#).

**API Request:** `https://{ip}/vmturbo/rest/login`

## Get Expected Initial Membership

After it logs in, the displays a list of virtual machines that reside on the host. These will be the group members.

The script uses the `/search` endpoint and retrieves the `EntityApiDTO` for the host. Using the `consumers` parameter, you can see each entity that purchases resources from this host. The script lists the consumers with a `className` that matches the entity type.

**Sample Request:** `GET https://{ip}/api/v3/search?q={host_name}&types=PhysicalMachine'&detail_type=entity'`

## Create the Filter Criterion

To create a dynamic group, you pass filter criteria to identify matching entities. You pass these criteria as an array. For this script, we pass a single criterion in the array.

**Example `criteria_list`:**

```
[
 {
 "expType": "RXEQ",
 "expVal": "host_name",
 "filterType": "vmsByPMName",
 "caseSensitive": "false"
 }
]
```

## Construct the input DTO and Create the Group

The script will take the `criterion` parameter and use it to construct the `GroupApiDTO`, with the following parameters:

- `isStatic`

When true, the created group is static. The script sets this parameter to false.

- `displayName`

The display name for the group. This name will appear in the user interface. The script sets this parameter

- `memberUuidList`

For a dynamic group, pass an empty array.

- `criteriaList`

The filtering criteria used to determine group membership. The script sets this parameter equal to the `criteria_list` variable.

- `groupType`

The entity type of the group members. The script sets this parameter equal to the `group_entity_type` variable.

**API Request:** POST `https://{ip}/api/v3/groups/`

The response to this request will be the `GroupApiDTO` for the created group.

#### Sample Response:

```
{
 "uuid": "285126581114848",
 "displayName": "TestGrp2",
 "className": "Group",
 "membersCount": 3,
 "entitiesCount": 3,
 "groupType": "VirtualMachine",
 "severity": "CRITICAL",
 "environmentType": "ONPREM",
 "isStatic": false,
 "logicalOperator": "AND",
 "criteriaList": [
 {
 "expVal": "hp-esx119.eng.vmturbo.com",
 "expType": "RXEQ",
 "filterType": "vmsByPMName",
 "caseSensitive": false,
 "entityType": null,
 "singleLine": false
 }
],
 "memberUuidList": [
 "73943582577376",
 "73943582577568",
 "73943582577744"
],
 "temporary": false,
 "activeEntitiesCount": 2,
 "cloudType": "UNKNOWN",
 "memberTypes": [
 "VirtualMachine"
],
 "entityTypes": [
 "VirtualMachine"
],
 "groupClassName": "GroupApiDTO"
}
```

## Verify the Group

To verify that the group was created successfully, the script uses the returned UUID to get the group data via the `/groups` endpoint. If the request gets the group, the membership of the group should match what was printed to the console in the *Get Expected Initial Membership* portion of the script, assuming that no virtual machines have been added or removed since that time.

If the group does not exist for the given UUID, the API returns the request status 404, with the message, `Group not found: <Passed_UUID>`. This indicates that the script failed to create the group.

**API Request:** `https://{ip}/api/v3/groups/{group_uuid}`

You might choose to only use a subset of the provided response. For example, this script prints the `displayName` and `UUID` for each group member.

### Sample Script Response:

```
Group Members:

Display name: DC26-P02-03
OID: 73943582577376
Display name: DC26-P02-02
OID: 73943582577568
Display name: DC26-P02-01
OID: 73943582577744
```

### Script Example

```
#
Commandline args: platformIP, username, pwd, group_name, entity_type, host_name
EXAMPLE: ./entity_changes.py 10.10.123.456 MyUser MyPwd MyTestGroup VirtualMachine MyHost
#
import json
import requests
import urllib3
import re

import sys
sys.path.append('../login') #This makes login visible...
import login

#
Make sure you have the correct count of arguments.
#
if len(sys.argv) != 7:
 print("Incorrect arguments list.")
 sys.exit()

#
Get the args into well named vars, and also authenticate.
#
ip = sys.argv[1]
group_name = sys.argv[4]
entity_type = sys.argv[5]
host_name = sys.argv[6]

token = login.get_cookie(ip, sys.argv[2], sys.argv[3])
authHeader = {'cookie': token}

#
Get Expected Initial Membership
#
response = requests.get(
 f'https://{ip}/api/v3/search?q={host_name}&types=PhysicalMachine'
 f'&detail_type=entity',
 headers=authHeader, verify=False)
get_vms_response_json = json.loads(response.text)

print(f'CONSUMERS: {get_vms_response_json[0]["consumers"]}')
print(f'The {entity_type} members of this dynamic group will be:')
for entity in get_vms_response_json[0]['consumers']:
 if entity['className'] == entity_type:
 print(entity)

#
Create the filter criterion
#
criterion = {}
criterion['expType'] = "RXEQ"
criterion['expVal'] = host_name
criterion['filterType'] = "vmsByPMName"
criterion['caseSensitive'] = "false"

#
Construct the Input DTO and Create the Group
#
```

```

group_input_dto = {}
group_input_dto['isStatic'] = 'false'
group_input_dto['displayName'] = group_name
group_input_dto['memberUuidList'] = []
group_input_dto['criterialist'] = [criterion]
group_input_dto['groupType'] = entity_type

response = requests.post(
 f'https://{ip}/api/v3/groups', headers=authHeader, json=group_input_dto, verify=False,
 stream=True)
group_creation_response_json = json.loads(response.text)
group_uuid = group_creation_response_json['uuid']

#
Verify the group
#
response = requests.get(
 f'https://{ip}/api/v3/groups/{group_uuid}/entities', headers=authHeader, verify=False)
group_members_response_json = json.loads(response.text)

Print a list of group members
print('\nGROUP MEMBERS:')
for item in group_members_response_json:
 print('Display name: ' + item['displayName'])
 print('OID: ' + item['uuid'])

```

## Reference: Internal Turbonomic API Requests

Internal Requests are requests that are used internally by Turbonomic, but are not intended for customer use. The structure data returned by these requests is subject to change without notice.

The following requests have been marked as Internal Only:

- /entities

```

DELETE /entities/loggingEntities
PUT /entities/{entity_Uuids}/loggingEntities
GET /entities/{entity_Uuid}/aspects/{aspect_name}/actions
POST /entities/{entity_Uuid}/aspects/{aspect_name}/actions
GET /entities/{entity_Uuid}/aspects/{aspect_name}/stats
POST /entities/{entity_Uuid}/aspects/{aspect_name}/stats
POST /entities/{entity_Uuid}/tags
DELETE /entities/{entity_Uuid}/tags
DELETE /entities/{entity_Uuid}/tags/{tag_key}

```

- /groups

```

GET /groups/{group_Uuid}/aspects/{aspect_name}/actions
POST /groups/{group_Uuid}/aspects/{aspect_name}/actions
GET /groups/{group_Uuid}/aspects/{aspect_name}/stats
POST /groups/{group_Uuid}/aspects/{aspect_name}/stats
POST /groups/{group_Uuid}/notifications/stats
POST /groups/{group_Uuid}/tags
DELETE /groups/{group_Uuid}/tags
DELETE /groups/{group_Uuid}/tags/{tag_key}

```

- /markets

```

PUT /markets/{market_Uuid}/policies/{policy_Uuid}

```

- /reports  
POST /reports/templates
- /scenarios  
PUT /scenarios/{scenario\_Uuid}  
PUT /scenarios/{scenario\_Uuid}/entities/{entity\_Uuid}  
DELETE /scenarios/{scenario\_Uuid}/entities/{entity\_Uuid}  
PUT /scenarios/{scenario\_Uuid}/groups/{group\_Uuid}  
PUT /scenarios/{scenario\_Uuid}/policies  
DELETE /scenarios/{scenario\_Uuid}/policies/{policy\_Uuid}  
PUT /scenarios/{scenario\_Uuid}/templates/{template\_name}
- /widgetsets  
The entire /widgetsets endpoint is Internal Only.

## Reference: Scenario Peak-of-Peak Full Scenario Input DTO

For reference purposes, see below for a complete input DTO for a Peak of Peaks plan.

```

 {
 "uuid": "XXXXDemo",
 "displayName": "XXXXDemo",
 "owners": [
 {
 "uuid": "_4T_7kwY-Ed-WUKbEYSVIDw",
 "username": "administrator",
 "showSharedUserSC": false
 }
],
 "type": "PEAKOFPEAKS",
 "scope": [
 {
 "uuid": "_PE0v-YEUee_hYfzgV9uYg",
 "displayName": "All On-prem Hosts",
 "className": "All On-prem Hosts"
 }
],
 "topologyChanges": {
 },
 "loadChanges": {
 "maxUtilizationList": [
 {
 "maxPercentage": 70,
 "projectionDay": 0,
 "target": {
 "displayName": "All On-prem Hosts",
 "className": "Group",
 "uuid": "_PE0v-YEUee_hYfzgV9uYg"
 }
 }
]
 },
 "overlayStatsList": [
 {
 "uuid": "4223ae0c-c99b-f6ca-c1ef-3f367bdf5528",
 "displayName": "centos_imp_2",
 "className": "VirtualMachine",
 "environmentType": "ONPREM",
 "stats": [
 {
 "displayName": "centos_imp_2",
 "date": "2019-02-01T00:00:00-05:00",
 "statistics": [
 {
 "displayName": "centos_imp_2",
 "date": "2019-03-14T15:12:36-04:00",
 "statistics": [

```

```

 "name": "priceIndex",
 "values": {
 "max": 20000,
 "min": 20000,
 "avg": 20000,
 "total": 20000
 },
 "value": 20000
 },
 {
 "name": "CPUProvisioned",
 "capacity": {
 "max": 104000,
 "min": 104000,
 "avg": 104000,
 "total": 104000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName": "hp-esx4.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "MHz",
 "values": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "value": 5200
 },
 {
 "name": "Mem",
 "capacity": {
 "max": 16776692,
 "min": 16776692,
 "avg": 16776692,
 "total": 16776692
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName": "hp-esx4.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "KB",
 "values": {
 "max": 3555836,
 "min": 3554799.8,
 "avg": 3554799.8,
 "total": 3554799.8
 },
 "value": 3554799.8
 },
 {
 "name": "MemProvisioned",
 "capacity": {
 "max": 167766912,
 "min": 167766912,
 "avg": 167766912,
 "total": 167766912
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName": "hp-esx4.dev.mycorp.com",

```

```

 "className": "PhysicalMachine"
 },
 "units": "KB",
 "values": {
 "max": 4194304,
 "min": 4194304,
 "avg": 4194304,
 "total": 4194304
 },
 "value": 4194304
},
{
 "name": "StorageAmount",
 "capacity": {
 "max": 3301375,
 "min": 3301375,
 "avg": 3301375,
 "total": 3301375
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "6f76b47e-e6f0a3fe",
 "displayName": "QSGRID01:ESXDC1DS1",
 "className": "Storage"
 },
 "units": "MB",
 "values": {
 "max": 725.32,
 "min": 725.32,
 "avg": 725.32,
 "total": 725.32
 },
 "value": 725.32
},
{
 "name": "NetThroughput",
 "capacity": {
 "max": 3072000,
 "min": 3072000,
 "avg": 3072000,
 "total": 3072000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName": "hp-esx4.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "Kbit/sec",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
},
{
 "name": "MemAllocation",
 "capacity": {
 "max": 52503552,
 "min": 52503552,
 "avg": 52503552,
 "total": 52503552
 },
 "filters": [
 {
 "type": "key",
 "value": "VirtualDataCenter::2332ba0f9dc14c07f11adb38e12df43c4bc8ca56"
 },
 {
 "type": "relation",

```



```

 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "2332ba0f9dc14c07f11adb38e12df43c4bc8ca56",
 "displayName": "olga_do_not_touch_1",
 "className": "VirtualDataCenter"
 },
 {
 "units": "KB",
 "values": {
 "max": 3555836,
 "min": 3554799.8,
 "avg": 3554799.8,
 "total": 3554799.8
 },
 "value": 3554799.8
 },
 {
 "name": "VCPU",
 "capacity": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "MHz",
 "values": {
 "max": 5214,
 "min": 5201.9,
 "avg": 5201.9,
 "total": 5201.9
 },
 "value": 5201.9
 },
 {
 "name": "Ballooning",
 "capacity": {
 "max": 16776692,
 "min": 16776692,
 "avg": 16776692,
 "total": 16776692
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName": "hp-esx4.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "KB",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
 },
 {
 "name": "StorageAccess",
 "capacity": {
 "max": 5000,
 "min": 5000,
 "avg": 5000,
 "total": 5000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 },
],

```

```

 "relatedEntity":{
 "uuid":"6f76b47e-e6f0a3fe",
 "displayName":"QSGRID01:ESXDC1DS1",
 "className":"Storage"
 },
 "units":"IOPS",
 "values":{
 "max":0,
 "min":0,
 "avg":0,
 "total":0
 },
 "value":0
 },
 {
 "name":"CPUAllocation",
 "capacity":{
 "max":31096,
 "min":31096,
 "avg":31096,
 "total":31096
 },
 "filters":[
 {
 "type":"key",
 "value":"VirtualDataCenter::2332ba0f9dc14c07f11adb38e12df43c4bc8ca56"
 },
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"2332ba0f9dc14c07f11adb38e12df43c4bc8ca56",
 "displayName":"olga_do_not_touch_1",
 "className":"VirtualDataCenter"
 },
 "units":"MHz",
 "values":{
 "max":5214,
 "min":5201.9,
 "avg":5201.9,
 "total":5201.9
 },
 "value":5201.9
 },
 {
 "name":"IOThroughput",
 "capacity":{
 "max":11264000,
 "min":11264000,
 "avg":11264000,
 "total":11264000
 },
 "reserved":{
 "max":9011200,
 "min":9011200,
 "avg":9011200,
 "total":9011200
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName":"hp-esx4.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"Kbit/sec",
 "values":{
 "max":0,
 "min":0,
 "avg":0,
 "total":0
 },
 "value":0
 },
 {
 "name":"StorageLatency",

```

```

 "capacity":{
 "max":100,
 "min":100,
 "avg":100,
 "total":100
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"6f76b47e-e6f0a3fe",
 "displayName":"QSGRID01:ESXDC1DS1",
 "className":"Storage"
 },
 "units":"msec",
 "values":{
 "max":0.5,
 "min":0.5,
 "avg":0.5,
 "total":0.5
 },
 "value":0.5
 },
 {
 "name":"VMem",
 "capacity":{
 "max":4194304,
 "min":4194304,
 "avg":4194304,
 "total":4194304
 },
 "filters":[
 {
 "type":"relation",
 "value":"sold"
 }
],
 "units":"KB",
 "values":{
 "max":3397384,
 "min":3233807,
 "avg":3233807,
 "total":3233807
 },
 "value":3233807
 },
 {
 "name":"CPU",
 "capacity":{
 "max":10400,
 "min":10400,
 "avg":10400,
 "total":10400
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238255f-86cc-4cb9-0f47-11ff689f0b22",
 "displayName":"hp-esx4.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"MHz",
 "values":{
 "max":5214,
 "min":5201.9,
 "avg":5201.9,
 "total":5201.9
 },
 "value":5201.9
 },
 {
 "name":"VStorage",
 "capacity":{
 "max":2646,
 "min":1014,

```



```

 "min":40000000,
 "avg":40000000,
 "total":40000000
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName":"hp-esx1.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"bit/sec",
 "values":{
 "max":0,
 "min":0,
 "avg":0,
 "total":0
 },
 "value":0
},
{
 "name":"NumDisk",
 "capacity":{
 "max":2147483650,
 "min":2147483650,
 "avg":2147483650,
 "total":2147483650
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName":"hp-esx1.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "values":{
 "max":1,
 "min":1,
 "avg":1,
 "total":1
 },
 "value":1
},
{
 "name":"Mem",
 "capacity":{
 "max":16776692,
 "min":16776692,
 "avg":16776692,
 "total":16776692
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName":"hp-esx1.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"KB",
 "values":{
 "max":3069332,
 "min":3062497,
 "avg":3062497,
 "total":3062497
 },
 "value":3062497
},
{
 "name":"MemProvisioned",
 "capacity":{

```

```

 "max":167766912,
 "min":167766912,
 "avg":167766912,
 "total":167766912
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName":"hp-esx1.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"KB",
 "values":{
 "max":4194304,
 "min":4194304,
 "avg":4194304,
 "total":4194304
 },
 "value":4194304
},
{
 "name":"StorageAmount",
 "capacity":{
 "max":3301375,
 "min":3301375,
 "avg":3301375,
 "total":3301375
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"6f76b47e-e6f0a3fe",
 "displayName":"QSGRID01:ESXDC1DS1",
 "className":"Storage"
 },
 "units":"MB",
 "values":{
 "max":719,
 "min":719,
 "avg":719,
 "total":719
 },
 "value":719
},
{
 "name":"NetThroughput",
 "capacity":{
 "max":3072000,
 "min":3072000,
 "avg":3072000,
 "total":3072000
 },
 "filters":[
 {
 "type":"relation",
 "value":"bought"
 }
],
 "relatedEntity":{
 "uuid":"Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName":"hp-esx1.dev.mycorp.com",
 "className":"PhysicalMachine"
 },
 "units":"Kbit/sec",
 "values":{
 "max":0,
 "min":0,
 "avg":0,
 "total":0
 },
 "value":0
},
}

```

```

"name": "MemAllocation",
"capacity": {
 "max": 52503552,
 "min": 52503552,
 "avg": 52503552,
 "total": 52503552
},
"filters": [
 {
 "type": "key",
 "value": "VirtualDataCenter::2332ba0f9dc14c07f11adb38e12df43c4bc8ca56"
 },
 {
 "type": "relation",
 "value": "bought"
 }
],
"relatedEntity": {
 "uuid": "2332ba0f9dc14c07f11adb38e12df43c4bc8ca56",
 "displayName": "olga_do_not_touch_1",
 "className": "VirtualDataCenter"
},
"units": "KB",
"values": {
 "max": 3069332,
 "min": 3062497,
 "avg": 3062497,
 "total": 3062497
},
"value": 3062497
},
{
 "name": "VCPU",
 "capacity": {
 "max": 5200,
 "min": 5200,
 "avg": 5200,
 "total": 5200
 },
 "filters": [
 {
 "type": "relation",
 "value": "sold"
 }
],
 "units": "MHz",
 "values": {
 "max": 5203,
 "min": 5194.67,
 "avg": 5194.67,
 "total": 5194.67
 },
 "value": 5194.67
},
{
 "name": "Ballooning",
 "capacity": {
 "max": 16776692,
 "min": 16776692,
 "avg": 16776692,
 "total": 16776692
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName": "hp-esx1.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "KB",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
},

```

```

{
 "name": "StorageAccess",
 "capacity": {
 "max": 5000,
 "min": 5000,
 "avg": 5000,
 "total": 5000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "6f76b47e-e6f0a3fe",
 "displayName": "QSGRID01:ESXDC1DS1",
 "className": "Storage"
 },
 "units": "IOPS",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
},
{
 "name": "CPUAllocation",
 "capacity": {
 "max": 31096,
 "min": 31096,
 "avg": 31096,
 "total": 31096
 },
 "filters": [
 {
 "type": "key",
 "value": "VirtualDataCenter::2332ba0f9dc14c07f11adb38e12df43c4bc8ca56"
 },
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "2332ba0f9dc14c07f11adb38e12df43c4bc8ca56",
 "displayName": "olga_do_not_touch_1",
 "className": "VirtualDataCenter"
 },
 "units": "MHz",
 "values": {
 "max": 5203,
 "min": 5194.67,
 "avg": 5194.67,
 "total": 5194.67
 },
 "value": 5194.67
},
{
 "name": "IOThroughput",
 "capacity": {
 "max": 11264000,
 "min": 11264000,
 "avg": 11264000,
 "total": 11264000
 },
 "reserved": {
 "max": 9011200,
 "min": 9011200,
 "avg": 9011200,
 "total": 9011200
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",

```



```

 "displayName": "hp-esx1.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 {
 "units": "Kbit/sec",
 "values": {
 "max": 0,
 "min": 0,
 "avg": 0,
 "total": 0
 },
 "value": 0
 },
 {
 "name": "StorageLatency",
 "capacity": {
 "max": 100,
 "min": 100,
 "avg": 100,
 "total": 100
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "6f76b47e-e6f0a3fe",
 "displayName": "QSGRID01:ESXDC1DS1",
 "className": "Storage"
 },
 "units": "msec",
 "values": {
 "max": 0.33,
 "min": 0.33,
 "avg": 0.33,
 "total": 0.33
 },
 "value": 0.33
 },
 {
 "name": "Q2VCPU",
 "capacity": {
 "max": 20000,
 "min": 20000,
 "avg": 20000,
 "total": 20000
 },
 "reserved": {
 "max": 10000,
 "min": 10000,
 "avg": 10000,
 "total": 10000
 },
 "filters": [
 {
 "type": "relation",
 "value": "bought"
 }
],
 "relatedEntity": {
 "uuid": "Virtual_ESX_4238a511-8593-4469-2751-46611e09b6a8",
 "displayName": "hp-esx1.dev.mycorp.com",
 "className": "PhysicalMachine"
 },
 "units": "msec",
 "values": {
 "max": 11,
 "min": 11,
 "avg": 11,
 "total": 11
 },
 "value": 11
 },
 {
 "name": "VMem",
 "capacity": {
 "max": 4194304,
 "min": 4194304,
 "avg": 4194304,
 "total": 4194304
 },

```



## Reference: Tables

---

### Reference Tables Overview

This section contains reference tables for common API parameters, separated by endpoint. Note that filter tables may contain other reference tables for completeness.

#### Action Reference Tables

Action Mode:

- **DISABLED:**

Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.

- **RECOMMENDED:**

Recommend the action so a user can execute it via the given hypervisor or by other external means

- **MANUAL:**

Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.

- **AUTOMATED:**

You have directed Turbonomic to execute the action automatically.

- **COLLECTION (UNUSED):**

Legacy action mode, no longer used.

Action State:

- **DISABLED:**

An action whose policy has been disabled.

- **RECOMMEND:**

An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.

- **PENDING\_ACCEPT:**

When the action mode is set to manual, a recommended action that has not been accepted or cleared.

- **ACCEPTED:**

An action that has been accepted, but is not yet in progress.

- **IN\_PROGRESS:**

An action that is being executed.

- **SUCCEEDED:**

An action that was accepted and completed successfully.

- **FAILED:**

An action that was accepted and did not complete successfully.

- **REJECTED:**

When the action mode is set to manual, an action that has been rejected by the user.

- **CLEARED:**

When the action mode is set to manual, an action that is no longer recommended by the market.

- **ACCOUNTING:**

For cloud entities, an action to resize in order to use a different instance template.

- **QUEUED:**

When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.

Action Type:

- **START:**

Start an entity, and add it to the Turbonomic market.

- **MOVE:**

Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.

- **SUSPEND:**

Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.

- **ADD PROVIDER:**

This is equivalent to the START action, but for storage entities.

- **CHANGE:**

Move a virtual machine's storage between datastores.

- **PROVISION:**

Add capacity to your environment.

- **RECONFIGURE:**

Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.

- **DELETE:**

This is equivalent to the SUSPEND action, but for storage entities.

- **RESERVE\_ON\_PM:**

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.

- **RESERVE\_ON\_DS:**

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.

- **RESIZE\_FOR EFFICIENCY:**

A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.

- **RESIZE\_FOR PERFORMANCE:**

A resize that increases the capacity of a commodity, such as vCPU, or vMEM.

Action Filter Criteria:

- **actionModeList**

- **DISABLED:**

Do not recommend or perform the action. When you disable an action, the user interface will no longer display actions of that type.

- **RECOMMENDED:**

Recommend the action so a user can execute it via the given hypervisor or by other external means

- MANUAL:  
Recommend the action, and provide the option to execute that action through the Turbonomic user interface or an /actions API request.
- AUTOMATED:  
You have directed Turbonomic to execute the action automatically.
- COLLECTION (**UNUSED**):  
Legacy action mode, no longer used.
- actionStateList
  - DISABLED:  
An action whose policy has been disabled.
  - RECOMMEND:  
An action that cannot be automated due to policy settings or must be executed outside the system, such as the purchase of hardware.
  - PENDING\_ACCEPT:  
When the action mode is set to manual, a recommended action that has not been accepted or cleared.
  - ACCEPTED:  
An action that has been accepted, but is not yet in progress.
  - IN\_PROGRESS:  
An action that is being executed.
  - SUCCEEDED:  
An action that was accepted and completed successfully.
  - FAILED:  
An action that was accepted and did not complete successfully.
  - REJECTED:  
When the action mode is set to manual, an action that has been rejected by the user.
  - CLEARED:  
When the action mode is set to manual, an action that is no longer recommended by the market.
  - ACCOUNTING:  
For cloud entities, an action to resize in order to use a different instance template.
  - QUEUED:  
When more than 10 actions are to be executed for a single target, Turbonomic will place the 11th and all subsequent actions into a QUEUED state, and they will execute as the previous actions complete.
- actionTypesList
  - START:  
Start an entity, and add it to the Turbonomic market.
  - MOVE:  
Move an entity from one provider to another. For example, moving a VM between hosts, or a datastore between disk arrays.
  - SUSPEND:  
Suspend the entity to exclude its capacity from your environment. Suspend does not remove the entity from your environment, but the market does not include the entity in its analysis. Based on

market conditions, analysis can recommend restarting suspended entities, which includes them in market analysis once again.

- ADD PROVIDER:

This is equivalent to the START action, but for storage entities.

- CHANGE:

Move a virtual machine's storage between datastores.

- PROVISION:

Add capacity to your environment.

- RECONFIGURE:

Analysis recommends this when a VM is not able to correct a misconfiguration via another type of action. For example, analysis can recommend a RECONFIGURE action when a placement policy requires a VM to move to a host on a particular cluster, but cross-cluster moves are not permitted.

- DELETE:

This is equivalent to the SUSPEND action, but for storage entities.

- RESERVE\_ON\_PM:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular host.

- RESERVE\_ON\_DS:

When using the reservation functionality, this is the action to reserve the resources of a VM on a particular datastore.

- RESIZE\_FOR\_EFFICIENCY:

A resize that reduces the capacity of a commodity, such as vCPU, or vMEM.

- RESIZE\_FOR\_PERFORMANCE:

A resize that increases the capacity of a commodity, such as vCPU, or vMEM.

- cleared

Default: false.

When true, actions that were removed from the action list before being executed will be returned.

- costType

- Saving

An action that will decrease cloud spending.

- Investment

An action that will increase cloud spending.

- cumulative

Default: false.

When true, shows the savings for each action in a scope as a cumulative total, rather than showing the savings for individual actions.

- environmentType

- ONPREM:

Returns only actions related to entities that are part of your on-premises environment.

- CLOUD:

Returns only actions related to entities that are part of your cloud environment.

- groupBy

You may group the actions returned by the following criteria: [actionModes, actionStates, actionTypes, risk, riskSeverity, riskSubCategory]

- hasReservedInstance

Default: false.

When true, only actions that are related to reserved instances will be returned.

- relatedEntityTypes

Returns only actions related to entities of these types within the scope. For example, if your scope is a virtual datacenter and you specify VirtualMachine as a related entity type, the request will include only those actions related to VMs in the specified virtual datacenter.

Choose from the following entity types: [ Datacenter, PhysicalMachine, VirtualMachine, Storage, Application, Chassis, DiskArray, IOModule, StorageControl, Switch, VirtualDataCenter, VPod, DPod, Container, Database, DatabaseServerContainer, LogicalPool ]

- riskSeverityList

Returns only actions of the given severities: [ Unknown, Normal, Minor, Major, Critical ]

- riskSubCategoryList

- returns only actions of the given subcategories: [ Performance Assurance, Efficiency Improvement, Prevention, Compliance ]

### Entity Reference Tables

Entity State:

- ACTIVE:

Entity is actively consuming resources.

- EVACUATED:

Entity is ready to suspend and will not participate in the market, but will not generate a SUSPEND action.

**Note:** This state replaces the SUSPEND state, and requires additional configuration to enable. Turbonomic does not recommend enabling this state.

- FAILOVER:

This entity is reserved for failover and will not participate in the market.

- IDLE:

Entity is powered off and not actively consuming market resources.

- LAUNCH:

Entity is starting.

- MAINTENANCE:

Entity is in maintenance mode.

- NOT\_MONITORED:

Entity is not currently monitored by Turbonomic.

- QUEUED:

Entity is in the process of changing from one state to another.

**Note:** This state is internal to Turbonomic and will not appear in the UI, but may appear in an API response if the call is executed at the precise moment the entity is changing state.

- RESOURCE\_ALLOCATION:

Cloud entity is being created.

- RESOURCE\_RELEASE:

Cloud Entity is being deleted.

- SUSPEND:

Entity is suspended or the guest OS is in sleep, standby, or suspended state. The entity is not removed from the environment, but will not be used in market decisions. Based on market conditions, suspended entities can be started, re-adding them to the market.

- SUSPEND\_PENDING:

Entity is making preparations to suspend, or is the next entity of its type to suspend.

- TERMINATE\_PENDING:

Entity is in the process of being deleted.

- UNKNOWN:

Turbonomic is unable to obtain the current state of the entity.

This occurs when the entity is reported to the hypervisor as having a state of DISCONNECTED, ORPHANED, or INVALID.

Entity Type:

- APPLICATION
- APPLICATION\_SERVER
- BUSINESS\_ACCOUNT
- BUSINESS\_UNIT
- CHASSIS
- CLOUD\_SERVICE
- CONTAINER
- CONTAINER\_POD
- DATABASE
- DATABASE\_SERVER
- DATACENTER
- DISK\_ARRAY
- DPOD
- IO\_MODULE
- LOAD\_BALANCER
- LOGICAL\_POOL
- NETWORK
- PHYSICAL\_MACHINE
- STORAGE
- STORAGE\_CONTROLLER
- SWITCH
- VIRTUAL\_APPLICATION
- VIRTUAL\_DATACENTER
- VIRTUAL\_MACHINE
- VPOD

## Reference: DTO Index

---

Data transfer objects (DTOs) are used to encapsulate and send data to the Turbonomic API. The following DTOs are used by Turbonomic:



**Note:**

The DTO Index is a work in progress, and included in a draft stage for reference. The information contained in the Index is accurate, but may not be complete.

## ActionApiDTO

### Description

Model to describe an Action

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### actionImpactID

- **type:** integer
- **format:** int64
- **description:** The ID for the action, which will persist across restarts

#### marketID

- **type:** integer
- **format:** int64
- **description:** The ID of the market for which the action was generated

#### createTime

- **type:** string
- **description:** Creation time

#### updateTime

- **type:** string
- **description:** Update time

#### clearTime

- **type:** string
- **description:** Execution time

#### actionType

- **type:** string
- **description:** Type
- **enum:** ['START', 'MOVE', 'SCALE', 'ALLOCATE', 'SUSPEND', 'PROVISION', 'RECONFIGURE', 'RESIZE', 'DELETE', 'RIGHT\_SIZE', 'BUY\_RI']

#### actionState

- **type:** string

- **description:** Action State
- **enum:** ['PENDING\_ACCEPT', 'ACCEPTED', 'REJECTED', 'PRE\_IN\_PROGRESS', 'POST\_IN\_PROGRESS', 'IN\_PROGRESS', 'SUCCEEDED', 'FAILED', 'RECOMMENDED', 'DISABLED', 'QUEUED', 'CLEARED', 'ACCOUNTING', 'READY', 'FAILING']

#### actionMode

- **type:** string
- **description:** Action Mode
- **enum:** ['DISABLED', 'RECOMMEND', 'EXTERNAL\_APPROVAL', 'MANUAL', 'AUTOMATIC', 'COLLECTION']

#### userName

- **type:** string
- **description:** Username of the user that accepted the action [DEPRECATED - Use: userInfo.userName or scheduleInfo.scheduleName in executionStatus.executorInfo]

#### details

- **type:** string
- **description:** A user-readable string describing the action

#### importance

- **type:** number
- **format:** float
- **description:** Numeric value that describes the priority of the action

#### target

- **description:** Target entity for an action. For example, the VM in a Resize Action, or the host for a VM move
- **ServiceEntityApiDTO**

#### currentEntity

- **description:** Current entity, such as the current host that a VM resides on for a VM move action
- **ServiceEntityApiDTO**

#### newEntity

- **description:** Destination entity, such as the host that a VM will move to for a VM move action
- **ServiceEntityApiDTO**

#### currentValue

- **type:** string
- **description:** Current value of a property, for example vMEM for a VM resize action

#### newValue

- **type:** string
- **description:** Calculated value to resize to, such as vMEM for a VM resize action

#### valueUnits

- **type:** string
- **description:** Units of the currentValue and newValue, such as KB for a VM vMEM resize action

#### resizeAttribute

- **type:** string
- **description:** The commodity attribute to be resized

- **enum:** ['CAPACITY', 'LIMIT', 'RESERVATION']

#### template

- **description:** Template used for the action, such as in a Cloud entity provision or move action
- **TemplateApiDTO**

#### risk

- **description:** Cause of the action
- **LogEntryApiDTO**

#### prerequisites

- **type:** array
- **description:** Description(s) of any prerequisite(s) that must occur before execution is possible. If this field is populated, the action would otherwise be executable but has been blocked by this prerequisite, and is displayed accordingly. If this field is not populated, no prerequisite has caused the action mode to be modified, and the action is displayed as normal.
- – **type:** string

#### prioritizedAutomationOpportunities

- **type:** array
- **description:** The Action is generated using the 'Prioritize Automation' setting, indicating that it is possible to have more savings if these opportunities are satisfied
- – **type:** string

#### stats

- **type:** array
- **description:** Statistics, such as Mem, vCPU, costPrice
- – **StatApiDTO**

#### progress

- **type:** integer
- **format:** int32
- **description:** Moved the 'progress' percentage value inside 'executionStatus', available when action detail level is 'EXECUTION' - [DEPRECATED - Use: executionStatus.progress]

#### currentLocation

- **description:** The region (DataCenter) where the current service entity is located, for cloud migration actions
- **DiscoveredEntityApiDTO**

#### newLocation

- **description:** The region, represented as a DataCenter entity, where the target service entity will be located for cloud migration actions
- **DiscoveredEntityApiDTO**

#### virtualDisks

- **type:** array
- **description:** Virtual Disk used for the action, for example in a storage move
- – **VirtualDiskApiDTO**

#### policy

- **description:** The policy associated with the action
- **PolicyApiDTO**

### **relatedSettingsPolicies**

- **type:** array
- **description:** The related settings policies associated with the action
- – [BaseApiDTO](#)

### **actionSchedule**

- **description:** The schedule associated with the action
- [ActionScheduleApiDTO](#)

### **compoundActions**

- **type:** array
- **description:** Property for compound move actions
- – [ActionApiDTO](#)

### **reservedInstance**

- **description:** The Reserved Instance associated with the action
- [ReservedInstanceApiDTO](#)

### **reservedInstancePlatform**

- **type:** string
- **description:** The platform of the recommended RI if the action has a supersaving

### **externalActionName**

- **type:** string
- **description:** The name under which this action appears in third-party software or external approval backend

### **externalActionUrl**

- **type:** string
- **description:** URL to third-party software showing this action. Suitable for external action approval

### **executionStatus**

- **description:** Status details of the execution of the Action. Include percentage progress, error messages and execution times.
- [ActionExecutionAuditApiDTO](#)

### **executionCharacteristics**

- **description:** Characteristics of the execution of the Action. Include disruptiveness and reversibility.
- [ActionExecutionCharacteristicApiDTO](#)

### **relatedActionsCountByType**

- **type:** object
- **description:** [BETA]: Map of related actions count by relationship type that have some relation to this Action
- **additionalProperties:** {'type': 'integer', 'format': 'int32'}

### **actionID**

- **type:** integer
- **format:** int64
- **description:** Id of the action

# ActionApiInputDTO

## Description

Model to describe a Filter for Actions

## Required Parameters

### startTime

- **type:** string
- **description:** Start time of the period to filter, based on Actions Creation Time

## Optional Parameters

### actionRelationTypeFilter

- **type:** array
- **description:** Filters the main list of actions based on the types of relationships they have to other actions.
- – **type:** string
  - **enum:** ['NONE', 'BLOCKED\_BY', 'BLOCKING', 'CAUSED\_BY', 'CAUSING']

### environmentType

- **type:** string
- **description:** Filter the actions by Environment Type
- **enum:** ['ONPREM', 'CLOUD']

### costType

- **type:** string
- **description:** Filter the actions by cost Type
- **enum:** ['SAVING', 'INVESTMENT']

### descriptionQuery

- **description:** This query will match the details field of the ActionApiDTO to return actions that match the specified expression.
- **QueryInputApiDTO**

### riskQuery

- **description:** This query will match the risk description field of the ActionApiDTO to return actions that match the specified expression.
- **QueryInputApiDTO**

### executionCharacteristics

- **description:** Characteristics of the execution of the action, including disruptiveness and reversibility.
- **ActionExecutionCharacteristicInputApiDTO**

### savingsAmountRange

- **description:** This field will match the costPrice statistic field of the ActionApiDTO to return actions that have a costPrice amount between minValue and maxValue, inclusive
- **RangeInputApiDTO**

### hasSchedule

- **type:** boolean

- **description:** If true, only actions that have an associated execution schedule will be returned.

#### **hasPrerequisites**

- **type:** boolean
- **description:** If true, only actions that have a prerequisite preventing execution of the action will be returned.

#### **endTime**

- **type:** string
- **description:** End time of the period to filter, based on Actions Creation Time

#### **cleared**

- **type:** boolean
- **description:** Filter by Actions Executed

#### **cumulative**

- **type:** boolean
- **description:** Show the action savings as cumulative values

#### **hasReservedInstance**

- **type:** boolean
- **description:** Filter the actions by whether having related reserved instance

#### **excludeNotifications**

- **type:** boolean
- **description:** Exclude the notifications from the result

#### **limitEntities**

- **type:** integer
- **format:** int32
- **description:** Limit number of entities in the result

#### **detailLevel**

- **type:** string
- **description:** Detail level of the Action, 'standard' is the default value
- **enum:** ['STANDARD', 'EXECUTION']

#### **relatedEntityTypes**

- **type:** array
- **example:** Selecting a VCD, and specifying ""related\_entities": ["VirtualMachine"] will include the actions for any VM in the given VDC
- **description:** Related Entity Types to include from the supply chain for this entity
- – **type:** string

#### **targetEntityTypes**

- **type:** array
- **example:** Specifying ""target\_entities": ["VirtualMachine"] will include the actions where the target entity is a VM.
- **description:** Filter the actions where the target entity of the action matches the provided list of entity types
- – **type:** string

### **relatedCloudServiceProviderIds**

- **type:** array
- **description:** Filter the actions by the related Cloud Service Providers.
- – **type:** string

### **groupBy**

- **type:** array
- **description:** Fields used to Group the Stats
- – **type:** string

### **instanceTypeFilter**

- **description:** Filter the actions based on a list of Instance Type Uuids.
- **InstanceTypeFilterApiDTO**

### **onDemandCostFilter**

- **description:** Filter the actions based on a on-demand cost range.
- **OnDemandCostFilterApiDTO**

### **actionTypeList**

- **type:** array
- – **type:** string
  - **enum:** ['NONE', 'START', 'MOVE', 'SCALE', 'ALLOCATE', 'SUSPEND', 'TERMINATE', 'SPAWN', 'ADD\_PROVIDER', 'CHANGE', 'REMOVE\_PROVIDER', 'PROVISION', 'RECONFIGURE', 'RESIZE', 'RESIZE\_CAPACITY', 'WARN', 'RECONFIGURE\_THRESHOLD', 'DELETE', 'RIGHT\_SIZE', 'RESERVE\_ON\_PM', 'RESERVE\_ON\_DS', 'RESIZE\_FOR\_EFFICIENCY', 'RESIZE\_FOR\_PERFORMANCE', 'CROSS\_TARGET\_MOVE', 'MOVE\_TOGETHER', 'ACTIVATE', 'DEACTIVATE', 'BUY\_RI']

### **actionStateList**

- **type:** array
- – **type:** string
  - **enum:** ['PENDING\_ACCEPT', 'ACCEPTED', 'REJECTED', 'PRE\_IN\_PROGRESS', 'POST\_IN\_PROGRESS', 'IN\_PROGRESS', 'SUCCEEDED', 'FAILED', 'RECOMMENDED', 'DISABLED', 'QUEUED', 'CLEARED', 'ACCOUNTING', 'READY', 'FAILING']

### **actionModeList**

- **type:** array
- – **type:** string
  - **enum:** ['DISABLED', 'RECOMMEND', 'EXTERNAL\_APPROVAL', 'MANUAL', 'AUTOMATIC', 'COLLECTION']

### **riskSeverityList**

- **type:** array
- – **type:** string

### **riskSubCategoryList**

- **type:** array
- – **type:** string

## ActionDetailsApiDTO

### Description

Parent class for action details. Implementing classes include: CloudResizeActionDetailsApiDTO, NoDetailsApiDTO, RIBuyActionDetailsApiDTO, CloudProvisionActionDetailsApiDTO, CloudSuspendActionDetailsApiDTO, OnPremResizeActionDetailsApiDTO and ReconfigureActionDetailsApiDTO.

### Required Parameters

#### actionDetailType

- **type:** string

#### actionType

- **type:** string

## ActionExecutionApiDTO

### Description

Model to describe an Action Execution.

### Required Parameters

#### acceptTime

- **type:** string
- **description:** Time when actions were accepted (in ISO8601 format - YYYY-MM-DDTHH:MM:SS)

#### actionIds

- **type:** array
- **description:** List of action IDs
- – **type:** string
- **maxItems:** 2147483647
- **minItems:** 1

#### id

- **type:** string
- **description:** Action Execution ID

#### skippedActions

- **type:** array
- **description:** List of actions that were skipped in execution
- – **SkippedActionApiDTO**

## ActionExecutionAuditApiDTO

### Description

Model to describe an Action Execution status.



## Required Parameters

### executionTime

- **type:** string
- **description:** Start of the execution time

### executorInfo

- **description:** Information about the executor
- **ExecutorInfoApiDTO**

### state

- **type:** string
- **description:** State of the execution
- **enum:** ['PENDING\_ACCEPT', 'ACCEPTED', 'REJECTED', 'PRE\_IN\_PROGRESS', 'POST\_IN\_PROGRESS', 'IN\_PROGRESS', 'SUCCEEDED', 'FAILED', 'RECOMMENDED', 'DISABLED', 'QUEUED', 'CLEARED', 'ACCOUNTING', 'READY', 'FAILING']

## Optional Parameters

### completionTime

- **type:** string
- **description:** End of the execution time

### message

- **type:** string
- **description:** Status message, if the Action failed, it will contains the Error details

### progress

- **type:** integer
- **format:** int32
- **description:** Action execution progress value, from 0 to 100

## ActionExecutionCharacteristicApiDTO

### Description

Model to describe action execution characteristics

### Optional Parameters

#### disruptiveness

- **type:** string
- **description:** whether action is disruptive.
- **enum:** ['DISRUPTIVE', 'NON\_DISRUPTIVE']

#### reversibility

- **type:** string
- **description:** whether action is reversible.
- **enum:** ['REVERSIBLE', 'IRREVERSIBLE']

## ActionExecutionCharacteristicInputApiDTO

### Description

Model to describe action execution characteristics

### Optional Parameters

#### disruptiveness

- **type:** string
- **description:** whether action is disruptive.
- **enum:** ['DISRUPTIVE', 'NON\_DISRUPTIVE']

#### reversibility

- **type:** string
- **description:** whether action is reversible.
- **enum:** ['REVERSIBLE', 'IRREVERSIBLE']

## ActionExecutionInputApiDto

### Description

Model to describe an Action Execution input.

### Required Parameters

#### actionIds

- **type:** array
- **description:** List of action IDs
- – **type:** string
- **maxItems:** 2147483647
- **minItems:** 1

## ActionResourceImpactStatApiInputDTO

### Description

Model to describe an input for Action Resource Impact Stats

### Required Parameters

#### actionResourceImpactStatList

- **type:** array
- **description:** List of Action Resource Impact Stat Inputs
- – ActionResourceImpactStatInput

### Optional Parameters

#### actionRelationTypeFilter

- **type:** array

- **description:** Filters the main list of actions based on the types of relationships they have to other actions.
- – **type:** string
- **enum:** ['NONE', 'BLOCKED\_BY', 'BLOCKING', 'CAUSED\_BY', 'CAUSING']

#### environmentType

- **type:** string
- **description:** Filter the actions by Environment Type
- **enum:** ['ONPREM', 'CLOUD']

#### costType

- **type:** string
- **description:** Filter the actions by cost Type
- **enum:** ['SAVING', 'INVESTMENT']

#### descriptionQuery

- **description:** This query will match the details field of the ActionApiDTO to return actions that match the specified expression.
- **QueryInputApiDTO**

#### riskQuery

- **description:** This query will match the risk description field of the ActionApiDTO to return actions that match the specified expression.
- **QueryInputApiDTO**

#### executionCharacteristics

- **description:** Characteristics of the execution of the action, including disruptiveness and reversibility.
- **ActionExecutionCharacteristicInputApiDTO**

#### savingsAmountRange

- **description:** This field will match the costPrice statistic field of the ActionApiDTO to return actions that have a costPrice amount between minValue and maxValue, inclusive
- **RangeInputApiDTO**

#### hasSchedule

- **type:** boolean
- **description:** If true, only actions that have an associated execution schedule will be returned.

#### hasPrerequisites

- **type:** boolean
- **description:** If true, only actions that have a prerequisite preventing execution of the action will be returned.

#### actionStateList

- **type:** array
- – **type:** string
- **enum:** ['PENDING\_ACCEPT', 'ACCEPTED', 'REJECTED', 'PRE\_IN\_PROGRESS', 'POST\_IN\_PROGRESS', 'IN\_PROGRESS', 'SUCCEEDED', 'FAILED', 'RECOMMENDED', 'DISABLED', 'QUEUED', 'CLEARED', 'ACCOUNTING', 'READY', 'FAILING']

#### actionModeList

- **type:** array
- – **type:** string

- **enum:** ['DISABLED', 'RECOMMEND', 'EXTERNAL\_APPROVAL', 'MANUAL', 'AUTOMATIC', 'COLLECTION']

#### **riskSeverityList**

- **type:** array
- – **type:** string

#### **riskSubCategoryList**

- **type:** array
- – **type:** string

## **ActionResourceImpactStatInput**

### **Description**

Model to describe an Action Resource Impact Stat Input

### **Required Parameters**

#### **actionType**

- **type:** string
- **description:** Action Type
- **enum:** ['NONE', 'START', 'MOVE', 'SCALE', 'ALLOCATE', 'SUSPEND', 'TERMINATE', 'SPAWN', 'ADD\_PROVIDER', 'CHANGE', 'REMOVE\_PROVIDER', 'PROVISION', 'RECONFIGURE', 'RESIZE', 'RESIZE\_CAPACITY', 'WARN', 'RECONFIGURE\_THRESHOLD', 'DELETE', 'RIGHT\_SIZE', 'RESERVE\_ON\_PM', 'RESERVE\_ON\_DS', 'RESIZE\_FOR\_EFFICIENCY', 'RESIZE\_FOR\_PERFORMANCE', 'CROSS\_TARGET\_MOVE', 'MOVE\_TOGETHER', 'ACTIVATE', 'DEACTIVATE', 'BUY\_RI']

#### **commodityType**

- **type:** string
- **example:** VCPU
- **description:** Commodity Type that could be impacted by the action. For example, VCPU or VMEM for a Resize VM action.
- **enum:** ['ACTION\_PERMIT', 'ACTIVE\_SESSIONS', 'APPLICATION', 'BALLOONING', 'BUFFER\_COMMODITY', 'CLUSTER', 'COLLECTION\_TIME', 'COOLING', 'CONNECTION', 'COUPON', 'CPU', 'CPU\_ALLOCATION', 'CPU\_PROVISIONED', 'CPU\_REQUEST\_ALLOCATION', 'CROSS\_CLOUD\_MOVE\_SVC', 'CROSS\_CLUSTER\_MOVE\_SVC', 'DATACENTER', 'DATASTORE', 'DB\_CACHE\_HIT\_RATE', 'DB\_MEM', 'DISK\_ARRAY\_ACCESS', 'DRS\_SEGMENTATION', 'DSPM\_ACCESS', 'EXTENT', 'FLOW', 'FLOW\_ALLOCATION', 'HEAP', 'HOST\_LUN\_ACCESS', 'IMAGE\_CPU', 'IMAGE\_MEM', 'IMAGE\_STORAGE', 'INSTANCE\_DISK\_SIZE', 'INSTANCE\_DISK\_TYPE', 'IO\_THROUGHPUT', 'LICENSE\_ACCESS', 'MEM', 'MEM\_ALLOCATION', 'MEM\_REQUEST\_ALLOCATION', 'MEM\_PROVISIONED', 'NET\_THROUGHPUT', 'NETWORK', 'NUM\_DISK', 'NUMBER\_CONSUMERS', 'NUM\_VCORE', 'POOL\_CPU', 'POOL\_MEM', 'POOL\_STORAGE', 'PORT\_CHANNEL', 'POWER', 'Q16\_VCPU', 'Q1\_VCPU', 'Q2\_VCPU', 'Q32\_VCPU', 'Q4\_VCPU', 'Q64\_VCPU', 'Q8\_VCPU', 'RESPONSE\_TIME', 'SAME\_CLUSTER\_MOVE\_SVC', 'SEGMENTATION', 'SLA\_COMMODITY', 'SOFTWARE\_LICENSE\_COMMODITY', 'SPACE', 'STORAGE\_ACCESS', 'STORAGE\_ALLOCATION', 'STORAGE\_AMOUNT', 'STORAGE\_CLUSTER', 'STORAGE\_LATENCY', 'STORAGE\_PROVISIONED', 'SWAPPING', 'THREADS', 'TRANSACTION', 'TRANSACTION\_LOG', 'VCPU', 'VCPU\_LIMIT\_QUOTA', 'VCPU\_REQUEST', 'VCPU\_REQUEST\_QUOTA', 'VDC', 'VMEM', 'VMEM\_LIMIT\_QUOTA', 'VMEM\_REQUEST', 'VMEM\_REQUEST\_QUOTA', 'VMPM\_ACCESS', 'VSTORAGE', 'ZONE', 'LICENSE\_COMMODITY', 'Q3\_VCPU', 'NUMBER\_CONSUMERS\_PM', 'Q6\_VCPU', 'Q7\_VCPU', 'QN\_VCPU', 'RIGHT\_SIZE\_SVC', 'RIGHT\_SIZE\_DOWN', 'MOVE', 'Q5\_VCPU', 'STORAGE', 'NUMBER\_CONSUMERS\_STORAGE', 'ACCESS', 'RIGHT\_SIZE\_UP', 'VAPP\_ACCESS', 'HOT\_STORAGE', 'HA\_COMMODITY', 'NETWORK\_POLICY', 'SERVICE\_LEVEL\_CLUSTER',

'PROCESSING\_UNITS', 'TENANCY\_ACCESS', 'TEMPLATE\_ACCESS', 'BURST\_BALANCE',  
'DESIRED\_COUPON', 'NETWORK\_INTERFACE\_COUNT', 'BICLIQUE', 'TAINT', 'LABEL', 'UNKNOWN']

#### **targetEntityType**

- **type:** string
- **description:** Target entity type for an action. For example, VirtualMachine for a resize virtual machine action
- **enum:** ['ApplicationComponent', 'VirtualMachine', 'Storage', 'DiskArray', 'Database', 'DatabaseServer', 'Container', 'LogicalPool']

## **ActionScheduleApiDTO**

### **Description**

Model to describe an Action Schedule.

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **nextOccurrence**

- **type:** string
- **description:** The time of the next occurrence represented in ISO8601 local time (YYYY-MM-DDTHH:MM:SS). For example, 2019-04-04T19:15.

#### **nextOccurrenceTimestamp**

- **type:** integer
- **format:** int64
- **description:** The time of the next occurrence of this schedule represented in Timestamp in milliseconds. For example, 1562100619000

#### **timeZoneId**

- **type:** string
- **description:** The timezone of the schedule

#### **mode**

- **type:** string
- **description:** The action mode affected by the schedule
- **enum:** ['DISABLED', 'RECOMMEND', 'EXTERNAL\_APPROVAL', 'MANUAL', 'AUTOMATIC', 'COLLECTION']

#### **acceptedByUserForMaintenanceWindow**

- **type:** boolean
- **description:** Boolean signifying acceptance or rejection of the action.

#### **userName**

- **type:** string
- **description:** The user who accepted the action for the maintenance window.

#### **remainingTimeActiveInMs**

- **type:** integer
- **format:** int64
- **description:** The amount of time that the schedule is going to be active at the time of call in milliseconds. This field will not be populated if the schedule is not currently active.

## **ActionScopesApiInputDTO**

### **Description**

Model to describe the request for Actions in multiple Scopes

### **Required Parameters**

#### **actionInput**

- **description:** List of requested actions
- [ActionApiInputDTO](#)

### **Optional Parameters**

#### **scopes**

- **type:** array
- **description:** List of uuid to use as scope
- – **type:** string

#### **relatedType**

- **type:** string
- **description:** Get the actions for the entity type related to the scopes

## **ActionScopesResourceImpactStatApiInputDTO**

### **Description**

Model to describe the request for Actions Resource Impacts in multiple Scopes

### **Required Parameters**

#### **actionResourceImpactStatApiInput**

- **description:** Requested action resource impact stats
- [ActionResourceImpactStatApiInputDTO](#)

#### **scopes**

- **type:** array
- **description:** List of uuid to use as scope
- – **type:** string

## ActiveDirectoryApiDTO

### Description

Model to describe an Active Directory. It contains domain name and server name which are global to Turbonomic server.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### domainName

- **type:** string
- **description:** The domain name of current active directory

#### loginProviderURI

- **type:** string
- **description:** The server name of current active directory

#### isSecure

- **type:** boolean
- **description:** The server name of current active directory

#### groups

- **type:** array
- **description:** Active directory groups defined in Ops Mgr for current active directory
- – [ActiveDirectoryGroupApiDTO](#)

## ActiveDirectoryGroupApiDTO

### Description

Model to describe an Active Directory Group. Active Directory groups are used to manage user authentication. You can specify role and scope for the group, and any member of the AD group can log into Turbonomic with those privileges.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**type**

- **type:** string
- **description:** The user type for current active directory group

**roleName**

- **type:** string
- **description:** The user role name of current active directory group

**scope**

- **type:** array
- **description:** The scope of current active directory group
- – **GroupApiDTO**

## AddObjectApiDTO

**Description**

Model to describe an add change in the scenario

**Optional Parameters****projectionDays**

- **type:** array
- **description:** List of projection days, indicate when to apply the changes
- – **type:** integer
- **format:** int32

**target**

- **description:** Target of the change, entity or group
- **BaseApiDTO**

**count**

- **type:** integer
- **format:** int32
- **description:** How many times to apply the add change

**targetEntityType**

- **type:** string
- **description:** Type of entity to add from group target

## AggregateCommodityFieldApiDTO

**Description****Inherits From**

This DTO inherits from [FieldApiDTO](#)



## Required Parameters

### aggregation

- **type:** string
- **description:** The type of aggregation.
- **readOnly:** True
- **enum:** ['TOTAL', 'AVERAGE']

### commodityAttribute

- **type:** string
- **description:** The commodity attribute.
- **readOnly:** True
- **enum:** ['USED', 'CAPACITY', 'PEAK', 'CURRENT\_UTILIZATION', 'WEIGHTED\_HISTORICAL\_UTILIZATION', 'PERCENTILE\_HISTORICAL\_UTILIZATION']

### commodityType

- **type:** string
- **description:** The commodity name.
- **readOnly:** True
- **enum:** ['ACTION\_PERMIT', 'ACTIVE\_SESSIONS', 'APPLICATION', 'BALLOONING', 'BUFFER\_COMMODITY', 'CLUSTER', 'COLLECTION\_TIME', 'COOLING', 'CONNECTION', 'COUPON', 'CPU', 'CPU\_ALLOCATION', 'CPU\_PROVISIONED', 'CPU\_REQUEST\_ALLOCATION', 'CROSS\_CLOUD\_MOVE\_SVC', 'CROSS\_CLUSTER\_MOVE\_SVC', 'DATACENTER', 'DATASTORE', 'DB\_CACHE\_HIT\_RATE', 'DB\_MEM', 'DISK\_ARRAY\_ACCESS', 'DRS\_SEGMENTATION', 'DSPM\_ACCESS', 'EXTENT', 'FLOW', 'FLOW\_ALLOCATION', 'HEAP', 'HOST\_LUN\_ACCESS', 'IMAGE\_CPU', 'IMAGE\_MEM', 'IMAGE\_STORAGE', 'INSTANCE\_DISK\_SIZE', 'INSTANCE\_DISK\_TYPE', 'IO\_THROUGHPUT', 'LICENSE\_ACCESS', 'MEM', 'MEM\_ALLOCATION', 'MEM\_REQUEST\_ALLOCATION', 'MEM\_PROVISIONED', 'NET\_THROUGHPUT', 'NETWORK', 'NUM\_DISK', 'NUMBER\_CONSUMERS', 'NUM\_VCPU', 'POOL\_CPU', 'POOL\_MEM', 'POOL\_STORAGE', 'PORT\_CHANNEL', 'POWER', 'Q16\_VCPU', 'Q1\_VCPU', 'Q2\_VCPU', 'Q32\_VCPU', 'Q4\_VCPU', 'Q64\_VCPU', 'Q8\_VCPU', 'RESPONSE\_TIME', 'SAME\_CLUSTER\_MOVE\_SVC', 'SEGMENTATION', 'SLA\_COMMODITY', 'SOFTWARE\_LICENSE\_COMMODITY', 'SPACE', 'STORAGE\_ACCESS', 'STORAGE\_ALLOCATION', 'STORAGE\_AMOUNT', 'STORAGE\_CLUSTER', 'STORAGE\_LATENCY', 'STORAGE\_PROVISIONED', 'SWAPPING', 'THREADS', 'TRANSACTION', 'TRANSACTION\_LOG', 'VCPU', 'VCPU\_LIMIT\_QUOTA', 'VCPU\_REQUEST', 'VCPU\_REQUEST\_QUOTA', 'VDC', 'VMEM', 'VMEM\_LIMIT\_QUOTA', 'VMEM\_REQUEST', 'VMEM\_REQUEST\_QUOTA', 'VMPM\_ACCESS', 'VSTORAGE', 'ZONE', 'LICENSE\_COMMODITY', 'Q3\_VCPU', 'NUMBER\_CONSUMERS\_PM', 'Q6\_VCPU', 'Q7\_VCPU', 'QN\_VCPU', 'RIGHT\_SIZE\_SVC', 'RIGHT\_SIZE\_DOWN', 'MOVE', 'Q5\_VCPU', 'STORAGE', 'NUMBER\_CONSUMERS\_STORAGE', 'ACCESS', 'RIGHT\_SIZE\_UP', 'VAPP\_ACCESS', 'HOT\_STORAGE', 'HA\_COMMODITY', 'NETWORK\_POLICY', 'SERVICE\_LEVEL\_CLUSTER', 'PROCESSING\_UNITS', 'TENANCY\_ACCESS', 'TEMPLATE\_ACCESS', 'BURST\_BALANCE', 'DESIRED\_COUPON', 'NETWORK\_INTERFACE\_COUNT', 'BICLIQUE', 'TAINT', 'LABEL', 'UNKNOWN']

## AggregatedHealthResponseDTO

### Description

A DTO for a single aggregated health response item.

### Required Parameters

#### healthState

- **type:** string

- **description:** The (aggregated) health state for the subcategory.
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

#### **numberOfItems**

- **type:** integer
- **format:** int32
- **description:** Number of entities whose state matches healthState.

#### **subcategory**

- **type:** string
- **description:** Health subcategory.

### **Optional Parameters**

#### **recommendations**

- **type:** array
- **description:** A list of recommendations specific for each problem (error) type.
- – **Recommendation**

## **AutomatedEntityDefinitionData**

### **Description**

### **Inherits From**

This DTO inherits from [IEntityDefinitionData](#)

### **Required Parameters**

#### **entityType**

- **type:** string
- **description:** Connected entity type
- **enum:** ['ApplicationComponent', 'BusinessTransaction', 'Container', 'ContainerPod', 'DatabaseServer', 'Service', 'VirtualMachine']

#### **namePrefix**

- **type:** string
- **description:** Naming prefix for automated entity definition

### **Optional Parameters**

#### **generationAndConnectionAttribute**

- **description:** Defined criteria of connected entities
- **FilterApiDTO**

## **BaseApiDTO**

### **Description**

Basic information of an entity

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

## BaseCommodityApiDTO

### Description

Describes a commodity

### Required Parameters

#### type

- **type:** string
- **description:** The type of the commodity
- **enum:** ['ACTION\_PERMIT', 'ACTIVE\_SESSIONS', 'APPLICATION', 'BALLOONING', 'BUFFER\_COMMODITY', 'CLUSTER', 'COLLECTION\_TIME', 'COOLING', 'CONNECTION', 'COUPON', 'CPU', 'CPU\_ALLOCATION', 'CPU\_PROVISIONED', 'CPU\_REQUEST\_ALLOCATION', 'CROSS\_CLOUD\_MOVE\_SVC', 'CROSS\_CLUSTER\_MOVE\_SVC', 'DATACENTER', 'DATASTORE', 'DB\_CACHE\_HIT\_RATE', 'DB\_MEM', 'DISK\_ARRAY\_ACCESS', 'DRS\_SEGMENTATION', 'DSPM\_ACCESS', 'EXTENT', 'FLOW', 'FLOW\_ALLOCATION', 'HEAP', 'HOST\_LUN\_ACCESS', 'IMAGE\_CPU', 'IMAGE\_MEM', 'IMAGE\_STORAGE', 'INSTANCE\_DISK\_SIZE', 'INSTANCE\_DISK\_TYPE', 'IO\_THROUGHPUT', 'LICENSE\_ACCESS', 'MEM', 'MEM\_ALLOCATION', 'MEM\_REQUEST\_ALLOCATION', 'MEM\_PROVISIONED', 'NET\_THROUGHPUT', 'NETWORK', 'NUM\_DISK', 'NUMBER\_CONSUMERS', 'NUM\_VCPU', 'POOL\_CPU', 'POOL\_MEM', 'POOL\_STORAGE', 'PORT\_CHANNEL', 'POWER', 'Q16\_VCPU', 'Q1\_VCPU', 'Q2\_VCPU', 'Q32\_VCPU', 'Q4\_VCPU', 'Q64\_VCPU', 'Q8\_VCPU', 'RESPONSE\_TIME', 'SAME\_CLUSTER\_MOVE\_SVC', 'SEGMENTATION', 'SLA\_COMMODITY', 'SOFTWARE\_LICENSE\_COMMODITY', 'SPACE', 'STORAGE\_ACCESS', 'STORAGE\_ALLOCATION', 'STORAGE\_AMOUNT', 'STORAGE\_CLUSTER', 'STORAGE\_LATENCY', 'STORAGE\_PROVISIONED', 'SWAPPING', 'THREADS', 'TRANSACTION', 'TRANSACTION\_LOG', 'VCPU', 'VCPU\_LIMIT\_QUOTA', 'VCPU\_REQUEST', 'VCPU\_REQUEST\_QUOTA', 'VDC', 'VMEM', 'VMEM\_LIMIT\_QUOTA', 'VMEM\_REQUEST', 'VMEM\_REQUEST\_QUOTA', 'VMPM\_ACCESS', 'VSTORAGE', 'ZONE', 'LICENSE\_COMMODITY', 'Q3\_VCPU', 'NUMBER\_CONSUMERS\_PM', 'Q6\_VCPU', 'Q7\_VCPU', 'QN\_VCPU', 'RIGHT\_SIZE\_SVC', 'RIGHT\_SIZE\_DOWN', 'MOVE', 'Q5\_VCPU', 'STORAGE', 'NUMBER\_CONSUMERS\_STORAGE', 'ACCESS', 'RIGHT\_SIZE\_UP', 'VAPP\_ACCESS', 'HOT\_STORAGE', 'HA\_COMMODITY', 'NETWORK\_POLICY', 'SERVICE\_LEVEL\_CLUSTER', 'PROCESSING\_UNITS', 'TENANCY\_ACCESS', 'TEMPLATE\_ACCESS', 'BURST\_BALANCE', 'DESIRED\_COUPON', 'NETWORK\_INTERFACE\_COUNT', 'BICLIQUE', 'TAINT', 'LABEL', 'UNKNOWN']

### Optional Parameters

#### key

- **type:** string
- **description:** The specific commodity

## units

- **type:** string
- **description:** The units in which the commodity is measured, if any

## BaseGroupApiDTO

### Description

Base model to describe a Group.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### membersCount

- **type:** integer
- **format:** int32
- **description:** Number of members of the Group

## BasicActionApiDTO

### Description

Model to tersely describe an action, providing only a set of basic fields

### Optional Parameters

#### actionType

- **type:** string
- **description:** Type
- **enum:** ['START', 'MOVE', 'SCALE', 'ALLOCATE', 'SUSPEND', 'PROVISION', 'RECONFIGURE', 'RESIZE', 'DELETE', 'RIGHT\_SIZE', 'BUY\_RI']

#### details

- **type:** string
- **description:** A user-readable string describing the action

#### target

- **description:** Target entity for an action. For example, the VM in a Resize Action, or the host for a VM move
- [DiscoveredEntityApiDTO](#)

#### currentEntity

- **description:** Current entity, such as the current host that a VM resides on for a VM move action

- **DiscoveredEntityApiDTO**

#### **newEntity**

- **description:** Destination entity, such as the host that a VM will move to for a VM move action
- **DiscoveredEntityApiDTO**

#### **actionID**

- **type:** integer
- **format:** int64
- **description:** Id of the action

## **BillingFamilyApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [GroupApiDTO](#)

### **Optional Parameters**

#### **masterAccountUuid**

- **type:** string
- **description:** The UUID of the master account.

#### **businessUnitApiDTOList**

- **type:** array
- **description:** List of business units, for all accounts in this billing family.
- – **BusinessUnitApiDTO**

## **BooleanConditionApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [ConditionApiDTO](#)

### **Required Parameters**

#### **value**

- **type:** boolean
- **description:** The value of the field must be equal to this boolean value
- **readOnly:** True

## **BooleanFieldValueApiDTO**

### **Description**

## Inherits From

This DTO inherits from [FieldValueApiDTO](#)

## Required Parameters

### value

- **type:** boolean
- **description:** The value.
- **readOnly:** True

## BulkActionRequestApiDTO

### Description

BulkActionRequest

### Optional Parameters

#### state

- **type:** string
- **description:** state
- **enum:** ['RUNNING', 'STOPPED', 'SUSPENDED', 'MAINTENANCE', 'FAILOVER', 'UNKNOWN', 'STARTING', 'STOPPING']

#### error

- **type:** string
- **description:** error

#### entityUUID

- **type:** string

## BulkActionRequestInputDTO

### Description

BulkActionInput

### Optional Parameters

#### action

- **type:** string
- **description:** action
- **enum:** ['START', 'STOP']

#### entityUuids

- **type:** array
- **description:** entityUuids
- – **type:** string

# BusinessUnitApiDTO

## Description

Model to describe the property of a business unit.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

### discount

- **type:** number
- **format:** float
- **description:** Entity discount percentage

### priceAdjustment

- **description:** Entity price adjustment percentage
- **PriceAdjustmentDTO**

### accountId

- **type:** string
- **description:** Account ID

### master

- **type:** boolean
- **description:** Flag which shows whether this account is master or not

### associatedTargetId

- **type:** integer
- **format:** int64
- **description:** Target associated with this account, if workloads related to this account are being discovered. If no workloads are discovered for this account, the associatedTargetId will be empty.

### budget

- **description:** Budget of the business unit, user only need to set value and units(monthly)
- **StatApiDTO**

### businessUnitType

- **type:** string

- **description:** Type of the business unit instance
- **enum:** ['BUDGET', 'DISCOUNT', 'PRICE\_ADJUSTMENT', 'DISCOVERED']

#### cloudType

- **type:** string
- **description:** Type of the cloud target which is related to current business account.
- **enum:** ['AWS', 'AZURE', 'GCP', 'UNKNOWN']

#### accountType

- **type:** string
- **description:** The category of the current cloud account.
- **enum:** ['STANDARD', 'GOVERNMENT\_US']

#### riSupported

- **type:** boolean
- **description:** Flag which shows whether this account is RI supported

#### childrenBusinessUnits

- **type:** array
- **description:** UUIDs of the discovered children business units that will share the budget or discounts configuration of this business unit.
- – **type:** string

#### targets

- **type:** array
- **description:** Targets associated with the business unit
- – **TargetApiDTO**

#### costPrice

- **type:** number
- **format:** float
- **description:** Cost price per Hour

#### severity

- **type:** string
- **description:** Calculated using the highest severity of the member entities
- **enum:** ['UNKNOWN', 'NORMAL', 'MINOR', 'MAJOR', 'CRITICAL']

#### membersCount

- **type:** integer
- **format:** int32
- **description:** Number of members of the business unit

#### memberType

- **type:** string
- **description:** The type of service entities comprising the business unit

#### related

- **description:** The rate card and the custom pricing information the business unit is using
- **BusinessUnitRelatedApiDTO**



### resourceGroupsCount

- **type:** integer
- **format:** int32
- **description:** The number of Resource Groups belonging to the business unit

### displayable

- **type:** boolean
- **description:** Whether business unit is displayable or not

### pricingIdentifiers

- **type:** object
- **description:** Pricing identifiers for the business unit
- **additionalProperties:** {'type': 'string'}

## BusinessUnitApiInputDTO

### Description

Model to describe the parameters for creating a business unit.

### Required Parameters

#### name

- **type:** string
- **description:** Name of the business unit

### Optional Parameters

#### budget

- **description:** Budget of the business unit, user only need to set value
- **StatApiDTO**

#### childrenBusinessUnits

- **type:** array
- **description:** UUIDs of the discovered children business units, that will share the budget or discounts configuration of this business unit.
- – **type:** string

#### discount

- **type:** number
- **format:** float
- **description:** Business unit enterprise-level discount percentage. It will come to effect if cloud service or template-level discounts are not configured.

#### priceAdjustment

- **description:** Business unit enterprise-level price adjustment percentage. It will come to effect if cloud service or template-level discounts are not configured.
- **PriceAdjustmentDTO**

#### targets

- **type:** array
- – **type:** string

## nameWithBusinessAccountType

- **type:** string

## BusinessUnitPriceAdjustmentApiDTO

### Description

Price adjustment values for cloud services and templates related to a business unit. The price adjustment configured on an user-defined business unit apply to cloud services and templates offered by the children business units.

### Optional Parameters

#### servicePriceAdjustments

- **type:** array
- **description:** Price adjustment information for the related cloud services and templates.
- – [CloudServicePriceAdjustmentApiDTO](#)

#### serviceDiscounts

- **type:** array
- **description:** Discount information for the related cloud services and templates
- – [CloudServicePriceAdjustmentApiDTO](#)

## BusinessUnitRelatedApiDTO

### Description

Model to describe relationship between a business unit and its related rate cards and price adjustments

### Optional Parameters

#### rateCard

- **description:** Rate Card related to this Business Unit.
- [BaseApiDTO](#)

#### priceAdjustment

- **description:** Price Adjustment related to this Business Unit.
- [BaseApiDTO](#)

## BusinessUserEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### sessions

- **type:** array
- **description:** Information about sessions to entities (VM or other) established by this BusinessUser.

- – BusinessUserSessionApiDTO

## BusinessUserSessionApiDTO

### Description

Session information for a connection established by BusinessUser to an entity (VM or other)

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### businessUserUuid

- **type:** string
- **description:** UUID of the connected BusinessUser

#### connectedEntityUuid

- **type:** string
- **description:** UUID of the entity that user has a connection to

#### duration

- **type:** integer
- **format:** int64
- **description:** Time elapsed since the session start in seconds- sessions are always timed

## ChangePasswordApiDTO

### Description

### Optional Parameters

#### oldPassword

- **type:** string

#### newPassword

- **type:** string

## ClientInputDTO

### Description

## Required Parameters

### name

- **type:** string
- **description:** Name of client

### supportedServices

- **type:** array
- **description:** List of supported services
- – **type:** string

## ClientNetworkSitesMetadataApiDTO

### Description

### Optional Parameters

#### id

- **type:** string
- **description:** id

#### name

- **type:** string
- **description:** name

## ClientNetworkTokenApiDTO

### Description

### Optional Parameters

#### id

- **type:** string
- **description:** id

#### tokenData

- **type:** string
- **description:** Token data

## ClientNetworkTokenInputDTO

### Description

### Optional Parameters

#### claimLimit

- **type:** integer
- **format:** int32
- **example:** 1

- **description:** Claim Limit: Number of times the token can be reused
- **minimum:** 1
- **maximum:** 999

#### expiration

- **type:** integer
- **format:** int32
- **example:** 15
- **description:** Expiration time (in minutes) to claim the token
- **minimum:** 1
- **maximum:** 9999

## ClientNetworkTokensMetadataApiDTO

### Description

### Optional Parameters

#### id

- **type:** string
- **description:** id

#### created

- **type:** string
- **description:** Created

#### claimsMade

- **type:** integer
- **format:** int32
- **description:** Claims Made

#### claimsRemaining

- **type:** integer
- **format:** int32
- **description:** Claims Remaining

#### claimExpiration

- **type:** string
- **description:** Claim Expiration

## ClientServiceApiDTO

### Description

### Optional Parameters

#### id

- **type:** string
- **description:** id

### supportedServices

- **type:** array
- **description:** Supported services
- – **type:** string

### secret

- **type:** string
- **description:** Secret

### name

- **type:** string
- **description:** Name of client

## CloudAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### zone

- **description:** The zone of the cloud entity
- **BaseApiDTO**

#### region

- **description:** The region of the cloud entity
- **BaseApiDTO**

#### businessAccount

- **description:** The BusinessAccount of the cloud entity
- **BaseApiDTO**

#### resourceGroup

- **description:** The ResourceGroup of the cloud entity
- **BaseApiDTO**

#### template

- **description:** The template used to generate the entity
- **BaseApiDTO**

#### billingType

- **type:** string
- **description:** The Cloud Billing type

#### tier

- **type:** string
- **description:** The Tier of the storage

### **riCoveragePercentage**

- **type:** number
- **format:** float
- **description:** The RI coverage percentage

### **riCoverage**

- **description:** Used value stores VM coupons in use, capacity stores coupon capacity
- **StatApiDTO**

### **onDemandCost**

- **type:** number
- **format:** float
- **description:** On-demand cost paid for workload per Hour (includes Compute and License on-demand costs)

### **enhancedNetworking**

- **type:** string
- **description:** The type of Enhanced Networking supported. If the entity is an AWS VM, this field will be populated by a value describing what kind of enhanced networking, if any, the VM is currently using. If the entity is not an AWS VM, this field is not relevant and will not be populated or displayed.
- **enum:** ['NONE', 'ENA']

### **enaActive**

- **type:** string
- **description:** Whether ENA is active. If the entity is an AWS VM, this field will be populated by a value describing whether ENA is currently in use or not. If the entity is not an AWS VM, this field is not relevant and will not be populated or displayed.
- **enum:** ['Active', 'Inactive']

### **virtualizationType**

- **type:** string
- **description:** The virtualization type. If the entity is an AWS VM, this field will be populated by a value describing what kind of virtualization type, if any, the VM is currently using. If the entity is not an AWS VM, this field is not relevant and will not be populated or displayed.
- **enum:** ['PVM', 'HVM']

### **architecture**

- **type:** string
- **description:** If the entity is an AWS VM, this field will be populated by a value describing what is the architecture of the VM. If the entity is not an AWS VM, this field is not relevant and will not be populated or displayed.
- **enum:** ['32-bit', '64-bit']

### **nvme**

- **type:** string
- **description:** NVMe driver support. If the entity is an AWS VM, this field will be true if NVMe driver installed. If the entity is not an AWS VM, this field is not relevant and will not be populated or displayed.
- **enum:** ['Installed', 'Not installed']

### **entityUptime**

- **description:** The entity uptime

- **EntityUptimeApiDTO**

#### tenancy

- **type:** string
- **description:** The tenancy type for VMs
- **enum:** ['DEFAULT', 'DEDICATED', 'HOST']

## CloudCommitmentAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### startTimeInMilliseconds

- **type:** integer
- **format:** int64
- **description:** The start time of the Cloud Commitment in UTC.

#### expirationTimeInMilliseconds

- **type:** integer
- **format:** int64
- **description:** The expiration time of the Cloud Commitment in UTC.

#### termInMilliseconds

- **type:** integer
- **format:** int64
- **description:** The term of the Cloud Commitment.

#### cloudCommitmentScopeType

- **type:** string
- **description:** The scope of the Cloud Commitment.
- **enum:** ['FamilyScoped', 'FamilyAndOSScoped', 'CloudServiceScoped']

#### cloudCommitmentScopeDTO

- **description:** The info about the scope of the cloud commitment.
- **CloudCommitmentScopeDTO**

#### cloudCommitmentCapacityApiDTO

- **description:** The capacity specific info of the cloud commitment.
- **CloudCommitmentCapacityApiDTO**

#### payment

- **type:** string
- **description:** Payment option of the cloud commitment.
- **enum:** ['ALL\_UPFRONT', 'PARTIAL\_UPFRONT', 'NO\_UPFRONT']



### **providerSpecificType**

- **type:** string
- **description:** The provider specific type.
- **enum:** ['SAVINGS\_PLAN', 'COMMITTED\_USE']

### **commitmentStatus**

- **type:** string
- **description:** The cloud commitment status.
- **enum:** ['Unknown', 'Pending', 'Active', 'Expired', 'Canceled']

## **CloudCommitmentCapacityApiDTO**

### **Description**

Model to describe cloud commitment capacity

### **Optional Parameters**

#### **instanceCapacity**

- **type:** integer
- **format:** int64
- **description:** The instance capacity of the cloud commitment.

#### **spendCapacity**

- **type:** number
- **format:** double
- **description:** The spend capacity of the cloud commitment.

#### **commoditiesBought**

- **type:** object
- **description:** The capacity for each type commodity bought by the commitment.
- **additionalProperties:** {'type': 'number', 'format': 'double'}

## **CloudCommitmentScopeDTO**

### **Description**

Model to describe cloud commitment scope.

### **Optional Parameters**

#### **cloudFamilyReferenceApiDTO**

- **description:** The family scoped type specific info of the cloud commitment. If null, indicates service scoped cloud commitment
- [CloudFamilyReferenceApiDTO](#)

## **CloudFamilyReferenceApiDTO**

### **Description**

Model describe a reference to a cloud family

## Optional Parameters

### family

- **type:** string
- **description:** The family of the Cloud Commitment.

### entityType

- **type:** string
- **description:** The entity type the Cloud Commitment applies to.
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

## CloudProvisionActionDetailsApiDTO

### Description

### Inherits From

This DTO inherits from [ActionDetailsApiDTO](#)

## Optional Parameters

### onDemandRate

- **type:** number
- **format:** float

### onDemandCost

- **type:** number
- **format:** float

### subType

- **type:** string

## CloudResizeActionDetailsApiDTO

### Description

### Inherits From

This DTO inherits from [ActionDetailsApiDTO](#)

## Optional Parameters

### onDemandRateBefore

- **type:** number
- **format:** float

### onDemandRateAfter

- **type:** number
- **format:** float

### onDemandCostBefore

- **type:** number
- **format:** float

### onDemandCostAfter

- **type:** number
- **format:** float

### riCoverageBefore

- **description:** RI coupons used and capacity of the target entity before action
- **StatApiDTO**

### riCoverageAfter

- **description:** RI coupons used and capacity of the target entity after action
- **StatApiDTO**

### entityUptime

- **description:** The entity uptime which was taken into account when generating the action
- **EntityUptimeApiDTO**

### subType

- **type:** string

## CloudServicePriceAdjustmentApiDTO

### Description

Price adjustments for cloud service and related templates specified for a certain pricing model.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### environmentType

- **type:** string

- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

#### discount

- **type:** number
- **format:** float
- **description:** Entity discount percentage

#### priceAdjustment

- **description:** Entity price adjustment percentage
- **PriceAdjustmentDTO**

#### pricingModel

- **type:** string
- **description:** Pricing model, 'On demand' for time-based, 'Reserved' for coupon-based, 'Spot' for customer bidding
- **enum:** ['ON\_DEMAND', 'RESERVED', 'SPOT']

#### templateDiscounts

- **type:** array
- **description:** Discount values for the templates offered by the cloud service under certain pricing model.
- – **TemplatePriceAdjustmentDTO**

#### templatePriceAdjustments

- **type:** array
- **description:** Price adjustment values for the templates offered by the cloud service under certain pricing model.
- – **TemplatePriceAdjustmentDTO**

## CloudServicePricingModelApiDTO

### Description

Cloud service + pricing model pair that is available for a business unit.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### environmentType

- **type:** string
- **description:** Environment type

- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

#### **discount**

- **type:** number
- **format:** float
- **description:** Entity discount percentage

#### **priceAdjustment**

- **description:** Entity price adjustment percentage
- **PriceAdjustmentDTO**

#### **pricingModel**

- **type:** string
- **description:** Pricing model, 'On demand' for time-based, 'Reserved' for coupon-based, 'Spot' for customer bidding
- **enum:** ['ON\_DEMAND', 'RESERVED', 'SPOT']

## **CloudSuspendActionDetailsApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [ActionDetailsApiDTO](#)

### **Optional Parameters**

#### **onDemandRate**

- **type:** number
- **format:** float

#### **onDemandCost**

- **type:** number
- **format:** float

#### **subType**

- **type:** string

## **CommodityFieldApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [FieldApiDTO](#)

### **Required Parameters**

#### **commodityAttribute**

- **type:** string
- **description:** The commodity attribute.

- **readOnly:** True
- **enum:** ['USED', 'CAPACITY', 'PEAK', 'CURRENT\_UTILIZATION', 'WEIGHTED\_HISTORICAL\_UTILIZATION', 'PERCENTILE\_HISTORICAL\_UTILIZATION']

### commodityType

- **type:** string
- **description:** The commodity name.
- **readOnly:** True
- **enum:** ['ACTION\_PERMIT', 'ACTIVE\_SESSIONS', 'APPLICATION', 'BALLOONING', 'BUFFER\_COMMODITY', 'CLUSTER', 'COLLECTION\_TIME', 'COOLING', 'CONNECTION', 'COUPON', 'CPU', 'CPU\_ALLOCATION', 'CPU\_PROVISIONED', 'CPU\_REQUEST\_ALLOCATION', 'CROSS\_CLOUD\_MOVE\_SVC', 'CROSS\_CLUSTER\_MOVE\_SVC', 'DATACENTER', 'DATASTORE', 'DB\_CACHE\_HIT\_RATE', 'DB\_MEM', 'DISK\_ARRAY\_ACCESS', 'DRS\_SEGMENTATION', 'DSPM\_ACCESS', 'EXTENT', 'FLOW', 'FLOW\_ALLOCATION', 'HEAP', 'HOST\_LUN\_ACCESS', 'IMAGE\_CPU', 'IMAGE\_MEM', 'IMAGE\_STORAGE', 'INSTANCE\_DISK\_SIZE', 'INSTANCE\_DISK\_TYPE', 'IO\_THROUGHPUT', 'LICENSE\_ACCESS', 'MEM', 'MEM\_ALLOCATION', 'MEM\_REQUEST\_ALLOCATION', 'MEM\_PROVISIONED', 'NET\_THROUGHPUT', 'NETWORK', 'NUM\_DISK', 'NUMBER\_CONSUMERS', 'NUM\_VCPU', 'POOL\_CPU', 'POOL\_MEM', 'POOL\_STORAGE', 'PORT\_CHANNEL', 'POWER', 'Q16\_VCPU', 'Q1\_VCPU', 'Q2\_VCPU', 'Q32\_VCPU', 'Q4\_VCPU', 'Q64\_VCPU', 'Q8\_VCPU', 'RESPONSE\_TIME', 'SAME\_CLUSTER\_MOVE\_SVC', 'SEGMENTATION', 'SLA\_COMMODITY', 'SOFTWARE\_LICENSE\_COMMODITY', 'SPACE', 'STORAGE\_ACCESS', 'STORAGE\_ALLOCATION', 'STORAGE\_AMOUNT', 'STORAGE\_CLUSTER', 'STORAGE\_LATENCY', 'STORAGE\_PROVISIONED', 'SWAPPING', 'THREADS', 'TRANSACTION', 'TRANSACTION\_LOG', 'VCPU', 'VCPU\_LIMIT\_QUOTA', 'VCPU\_REQUEST', 'VCPU\_REQUEST\_QUOTA', 'VDC', 'VMEM', 'VMEM\_LIMIT\_QUOTA', 'VMEM\_REQUEST', 'VMEM\_REQUEST\_QUOTA', 'VMPM\_ACCESS', 'VSTORAGE', 'ZONE', 'LICENSE\_COMMODITY', 'Q3\_VCPU', 'NUMBER\_CONSUMERS\_PM', 'Q6\_VCPU', 'Q7\_VCPU', 'QN\_VCPU', 'RIGHT\_SIZE\_SVC', 'RIGHT\_SIZE\_DOWN', 'MOVE', 'Q5\_VCPU', 'STORAGE', 'NUMBER\_CONSUMERS\_STORAGE', 'ACCESS', 'RIGHT\_SIZE\_UP', 'VAPP\_ACCESS', 'HOT\_STORAGE', 'HA\_COMMODITY', 'NETWORK\_POLICY', 'SERVICE\_LEVEL\_CLUSTER', 'PROCESSING\_UNITS', 'TENANCY\_ACCESS', 'TEMPLATE\_ACCESS', 'BURST\_BALANCE', 'DESIRED\_COUPON', 'NETWORK\_INTERFACE\_COUNT', 'BICLIQUE', 'TAINT', 'LABEL', 'UNKNOWN']

## ComputeTierAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Required Parameters

#### instanceStorageSize

- **type:** number
- **format:** float
- **description:** Size per instance storage in GB. This will be 0 if no instance storage's are present for compute tier.

#### numInstanceStorages

- **type:** number
- **format:** float
- **description:** Number of instance storage disks. Compute tiers can have 0 or more instance storage disks.

### tierFamily

- **type:** string
- **description:** Family of compute tier (for example 'M4' for M4.large)

## ConditionApiDTO

### Description

A condition on a field of an entity or group.

### Required Parameters

#### conditionType

- **type:** string
- **description:** Type of compared values.
- **readOnly:** True
- **enum:** ['NUMBER', 'TEXT', 'INCLUSION', 'INTEGER', 'BOOLEAN']

#### field

- **description:** Field to compare.
- **readOnly:** True
- **FieldApiDTO**

#### operator

- **type:** string
- **description:** Operator to use.
- **readOnly:** True
- **enum:** ['EQ', 'NEQ', 'GT', 'LT', 'GE', 'LE', 'LIKE', 'IN']

## ConfigChangesApiDTO

### Description

Policies and Settings changes in the scenario

### Optional Parameters

#### addPolicyList

- **type:** array
- **description:** List of Placement Policies to add
- – **PolicyApiDTO**

#### removePolicyList

- **type:** array
- **description:** List of Placement Policies to remove
- – **PolicyApiDTO**

#### automationSettingList

- **type:** array
- **description:** List of automation settings
- – **SettingApiDTOString**

### **removeConstraintList**

- **type:** array
- **description:** List of constraints to remove
- – [RemoveConstraintApiDTO](#)

### **riSettingList**

- **type:** array
- **description:** List of Reserved Instance settings
- – [SettingApiDTO](#)

### **osMigrationSettingList**

- **type:** array
- **description:** List of OS migration settings
- – [SettingApiDTOSTring](#)

### **includedCoupons**

- **description:** List of Included Coupons (RIs) and related information
- [IncludedCouponsApiDTO](#)

### **subscription**

- **description:** Subscription
- [BusinessUnitApiDTO](#)

## **ConstraintApiDTO**

### **Description**

Model to describe the property of constraints, corresponding to entities constraints view in ui.

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **relation**

- **type:** string
- **enum:** ['sold', 'bought']

#### **entityType**

- **type:** string
- **description:** entity type of providers or consumers

#### **relatedEntities**

- **type:** array



- **description:** if relation is sold they are consumers, if bought they are providers

- – [ServiceEntityApiDTO](#)

#### numPotentialEntities

- **type:** integer
- **format:** int32
- **description:** how many other providers/consumers can fulfill the constraint

#### placementOptions

- **type:** array
- **description:** A list of constraints prevent finding better placement for an entity
- – [PlacementOptionApiDTO](#)

## ConstraintApiInputDTO

### Description

Model to describe the property of constraints

### Optional Parameters

#### relation

- **type:** string
- **enum:** ['sold', 'bought']

#### placementOptions

- **type:** array
- **description:** A list of constraints prevent finding better placement for an entity
- – [PlacementOptionApiInputDTO](#)

#### entityTypeFilter

- **type:** string
- **description:** potential entity type filter
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

## ContainerPlatformContextAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

## Optional Parameters

### namespace

- **type:** string
- **description:** The namespace that a container platform entity belong to - [DEPRECATED - Use: namespaceEntity]

### containerPlatformCluster

- **type:** string
- **description:** The container platform cluster name that an entity belongs to - [DEPRECATED - Use: containerClusterEntity]

### workloadControllerEntity

- **description:** The workload controller entity that an entity belongs to
- **BaseApiDTO**

### namespaceEntity

- **description:** The namespace entity that an entity belongs to
- **BaseApiDTO**

### containerClusterEntity

- **description:** The container platform cluster entity that an entity belongs to
- **BaseApiDTO**

## CostInputApiDTO

### Description

Model to describe the request for Cost Statistics by a Time range

## Optional Parameters

### tagFilters

- **type:** array
- **description:** List of Tag Filters. NOTE: Required non-empty when used with costGroupBys set to "TAG"
- – **TagApiDTO**

### costGroupBys

- **type:** array
- **description:** List of values to group the results by. NOTE: When used non-empty tagFilters required
- – **type:** string
  - **enum:** ['TAG', 'TAG\_GROUP', 'ENTITY', 'ENTITY\_TYPE', 'ACCOUNT', 'REGION', 'CLOUD\_SERVICE', 'SERVICE\_PROVIDER', 'CLOUD\_TIER\_PROVIDER', 'COST\_CATEGORY', 'PRICE\_MODEL']

### startDate

- **type:** string
- **description:** Start time of the period to filter. If startDate and endDate not provided current stats will be processed

### endDate

- **type:** string

- **description:** End time of the period to filter. If startDate and endDate not provided current stats will be processed

## CpuModelApiDTO

### Description

Model to describe a Template: contains the resources used to Deploy a VM or to Add Workload/Supply in a Plan

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### numCores

- **type:** integer
- **format:** int32
- **description:** Number of Cores

#### speed

- **type:** integer
- **format:** int32
- **description:** Speed

#### scalingFactor

- **type:** number
- **format:** double
- **description:** Scaling Factor

#### modelName

- **type:** string

## CriteriaOptionApiDTO

### Description

Generic model used to describe a search criteria option

### Optional Parameters

#### value

- **type:** string
- **description:** value

**displayName**

- **type:** string
- **description:** value to display, more readable

**subValues**

- **type:** array
- **description:** possible sub-values associated with the main value
- – **type:** string

## DBEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

**dbEngine**

- **type:** string
- **description:** The Database engine

**dbEdition**

- **type:** string
- **description:** The Database edition

**dbVersion**

- **type:** string
- **description:** The Database engine version

**licenseModel**

- **type:** string
- **description:** The license model of the database engine

**deploymentType**

- **type:** string
- **description:** The deployment type of the database engine

**maxConcurrentWorkers**

- **type:** integer
- **format:** int32
- **description:** Maximum concurrent workers allowed for the DB

**maxConcurrentSessions**

- **type:** integer
- **format:** int32
- **description:** Maximum concurrent sessions allowed for the DB

**pricingModel**

- **type:** string

- **example:** DTU
- **description:** Database Pricing Model.
- **enum:** ['DTU', 'vCore', 'DWU', 'RU']

#### replicationRole

- **type:** string
- **example:** Primary
- **description:** Replication role if DB is part of geo replicated DB. Currently applies to Azure DB only.
- **enum:** ['Primary', 'Secondary', 'None']

#### storageTier

- **type:** string
- **example:** GP2
- **description:** Storage Tier

#### dbServerName

- **type:** string
- **description:** The server name of a database

#### serviceTier

- **type:** string
- **example:** Hyperscale
- **description:** The service tier of a database, generally defines the storage architecture, space and I/O limits, and business continuity options related to availability and disaster recovery.
- **enum:** ['Free', 'Basic', 'Standard', 'Premium', 'GeneralPurpose', 'BusinessCritical', 'Hyperscale']

#### computeTier

- **type:** string
- **example:** Provisioned Compute
- **description:** The compute tier of a database. Compute tier options include the provisioned and serverless compute tiers

#### hardwareGeneration

- **type:** string
- **example:** Gen4
- **description:** Database hardware generation. The generation of hardware that underlies the DB instance

#### backupPolicy

- **type:** string
- **example:** Enabled
- **description:** The policy to backup and restore a database.

#### workspace

- **type:** string
- **description:** The resource group where the database is located.

#### idleTime

- **type:** integer
- **format:** int64
- **description:** Time duration for a database being idle in milliseconds.

### **poolType**

- **type:** string
- **description:** Type of database pool.

### **resourceId**

- **type:** string
- **description:** Resource identifier of the database. For example, the resource id in Azure, the ARN in AWS, or the full resource name in GCP.

### **serverConfiguration**

- **type:** string
- **description:** The Database server configuration

## **DatabaseServerEntityAspectApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [EntityAspect](#)

### **Optional Parameters**

#### **dbEngine**

- **type:** string
- **description:** The Database engine

#### **dbEdition**

- **type:** string
- **description:** The Database edition

#### **dbVersion**

- **type:** string
- **description:** The Database engine version

#### **licenseModel**

- **type:** string
- **description:** The license model of the database engine

#### **deploymentType**

- **type:** string
- **description:** The deployment type of the database engine

#### **maxConcurrentWorkers**

- **type:** integer
- **format:** int32
- **description:** Maximum concurrent workers allowed for the DB

#### **maxConcurrentSessions**

- **type:** integer
- **format:** int32

- **description:** Maximum concurrent sessions allowed for the DB

#### pricingModel

- **type:** string
- **example:** DTU
- **description:** Pricing Model followed by DB.

#### storageTier

- **type:** string
- **example:** GP2
- **description:** Storage Tier

#### clusterRole

- **type:** string
- **description:** Cluster Role
- **enum:** ['Writer', 'Reader']

#### hourlyBilledOps

- **type:** number
- **format:** double
- **description:** Hourly Billed Ops - Count of operations per hour

#### featureStateMap

- **type:** object
- **description:** Features
- **additionalProperties:** {'type': 'string', 'enum': ['Enabled', 'Disabled', 'Unknown']}

#### resourceId

- **type:** string
- **description:** Resource identifier of the database server. For example, the resource id in Azure, the ARN in AWS, or the full resource name in GCP.

#### creationTimeStamp

- **type:** integer
- **format:** int64
- **description:** Timestamp in milliseconds for the creation of the database server.

#### serverConfiguration

- **type:** string
- **description:** The Database server configuration

## DatabaseServerTierAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

## Required Parameters

### tierFamily

- **type:** string
- **description:** Family of database server tier (for example 'M5' for db.M5.large)

## DatabaseTierAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

## Required Parameters

### tierFamily

- **type:** string
- **description:** Family of database tier (for example 'S' for S4)

## DemandEntityInfoDTO

### Description

Model to describe a reservation entity

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### template

- [BaseApiDTO](#)

### deploymentProfile

- [BaseApiDTO](#)

### operationalCost

- **type:** string

### placements

- [PlacementInfoDTO](#)



# DemandReservationApiDTO

## Description

Basic information of an entity

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### prefix

- **type:** string

### count

- **type:** integer
- **format:** int32

### status

- **type:** string

### reserveDateTime

- **type:** string

### expireDateTime

- **type:** string

### deployDateTime

- **type:** string

### reserveCount

- **type:** integer
- **format:** int32

### deployCount

- **type:** integer
- **format:** int32

### placementExpirationDateTime

- **type:** string

### placementResultMessage

- **type:** string
- **description:** Details of placement during failure or success

### constraintInfos

- **type:** array

- **description:** The constraints user specified when running reservation which limit where templates could be placed.
- – **ReservationConstraintApiDTO**

#### reservationDeployed

- **type:** boolean
- **description:** When true, the reservation will be included in the system and will keep utilizing the resources from host and storage until the corresponding virtual machine utilization starts to reflect on the historically computed utilization.

#### mode

- **type:** string
- **description:** Demand related behavior for reservation of NO\_GROUPING/AFFINITY. This value interacts with the grouping value which is specifying the type of grouping that the demand should consider the mode for. NO\_GROUPING and NONE must be paired together and cannot be paired with other values. Combinations such as: (NO\_GROUPING - NONE), (AFFINITY - CLUSTER) are valid. Examples of invalid combinations are (NO\_GROUPING - CLUSTER), (AFFINITY - NONE).
- **enum:** ['NO\_GROUPING', 'AFFINITY']

#### grouping

- **type:** string
- **description:** Grouping related behavior for reservation of NONE and CLUSTER. This value interacts with the mode value.
- **enum:** ['NONE', 'CLUSTER']

#### scope

- **type:** array
- **description:** List of uuids of groups (currently limited to Clusters or Host Groups) whose members are the only possible providers of that type of entity. These are further constrained by the constraintIds passed in placement parameters. If multiple groups of providers are specified then it is possible that entities will be placed within different groups.
- – **type:** string

#### demandEntities

- **type:** array
- – **DemandEntityInfoDTO**

## DemandReservationApiInputDTO

### Description

### Required Parameters

#### action

- **type:** string
- **description:** Action requested for the reservation
- **enum:** ['RESERVATION']

#### demandName

- **type:** string
- **description:** Name of the reservation

- **minLength:** 1
- **maxLength:** 2147483647

#### parameters

- **type:** array
- **description:** Reservation Parameters
- – **DemandReservationParametersDTO**
- **maxItems:** 1
- **minItems:** 1

### Optional Parameters

#### mode

- **type:** string
- **description:** Demand related behavior for reservation of NO\_GROUPING/AFFINITY. This value interacts with the grouping value which is specifying the type of grouping that the demand should consider the mode for. NO\_GROUPING and NONE must be paired together and cannot be paired with other values. Combinations such as: (NO\_GROUPING - NONE), (AFFINITY - CLUSTER) are valid. Examples of invalid combinations are (NO\_GROUPING - CLUSTER), (AFFINITY - NONE).
- **enum:** ['NO\_GROUPING', 'AFFINITY']

#### grouping

- **type:** string
- **description:** Grouping related behavior for reservation of NONE and CLUSTER. This value interacts with the mode value.
- **enum:** ['NONE', 'CLUSTER']

#### reserveDateTime

- **type:** string
- **description:** Date to specify when to reserve the Demand

#### expireDateTime

- **type:** string
- **description:** Date to specify when the Demand request will expire, required if action = RESERVATION

#### deployDateTime

- **type:** string
- **description:** Date to specify when to deploy the Demand

#### scope

- **type:** array
- **description:** List of uuids of groups (currently limited to Clusters or Host Groups) whose members are the only possible providers of that type of entity. These are further constrained by the constraintIds passed in placement parameters. If multiple groups of providers are specified then it is possible that entities will be placed within different groups.
- – **type:** string

## DemandReservationParametersDTO

### Description

## Required Parameters

### placementParameters

- **description:** Placement parameters
- **PlacementParametersDTO**

## Optional Parameters

### deploymentParameters

- **description:** Deployment parameters, required if action = DEPLOYMENT, RESERVATION
- **DeploymentParametersDTO**

## DeploymentParametersDTO

### Description

### Optional Parameters

#### deploymentProfileID

- **type:** string
- **description:** Deployment Profile uuid

#### highAvailability

- **type:** boolean
- **description:** High availability

#### priority

- **type:** string
- **description:** Workload priority
- **enum:** ['LOW', 'NORMAL', 'MISSION\_CRITICAL']

## DeploymentProfileApiDTO

### Description

Model to describe a Deployment Profile, including target and provider entity parameters

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### account

- **description:** Business account related to the Deployment Profile

- [BaseApiDTO](#)

#### deployParameters

- **type:** array
- **description:** Target and Provider parameters of the Deployment Profile
- – [DeploymentProfileTargetApiDTO](#)

## DeploymentProfileParamApiDTO

### Description

Provider-specific Deployment Profile parameters

### Optional Parameters

#### parameterType

- **type:** string
- **description:** Parameter type. For vCenter, this can be a folder, osConfiguration, or image. For all other technologies, this must be an image.
- **enum:** ['image', 'folder', 'osConfiguration']

#### properties

- **type:** array
- **description:** A NameValueInputDTO describing the name and value of the parameter.
- – [NameValueInputDTO](#)

## DeploymentProfileProviderApiDTO

### Description

Deployment profile Provider properties

### Optional Parameters

#### provider

- **description:** Provider UUID
- [BaseApiDTO](#)

#### parameters

- **type:** array
- **description:** A DeploymentProfileParamApiDTO describing the parameters for the provider entities
- – [DeploymentProfileParamApiDTO](#)

## DeploymentProfileTargetApiDTO

### Description

Deployment Profile Target properties

## Optional Parameters

### targetType

- **type:** string
- **description:** Target type for the deployment profile
- **enum:** ['vCenter', 'AWS', 'Azure', 'Softlayer']

### providers

- **type:** array
- **description:** DeploymentProfileProviderApiInputDTO describing the provider entity parameters
- – **DeploymentProfileProviderApiDTO**

## DesktopPoolEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

## Optional Parameters

### cloneType

- **type:** string
- **description:** Type of a clone provided by the DesktopPool
- **enum:** ['FULL', 'LINKED', 'INSTANT']

### provisionType

- **type:** string
- **description:** How clones are provisioned in the DesktopPool
- **enum:** ['UPFRONT', 'ON\_DEMAND']

### assignmentType

- **type:** string
- **description:** Whether the same clone is assigned each time when user logs back in
- **enum:** ['PERSISTENT', 'DYNAMIC']

### masterTemplateUuid

- **type:** string
- **description:** UUID of a master template that is used to create clones. Only one of master VM and master template can be set.

### masterVirtualMachineUuid

- **type:** string
- **description:** UUID of a master virtual machine that is used to create clones. Only one of master VM and master template can be set.

### masterVirtualMachineSnapshot

- **type:** string
- **description:** Name of the snapshot of a master virtual machine that is used to create clones.

### **getvCenterClusterName**

- **type:** string
- **description:** Name of the vCenter cluster that this DesktopPool draws resources from.

### **vendorId**

- **type:** string
- **description:** Vendor Identifier for the DesktopPool.

## **DetailDataApiDTO**

### **Description**

A single entity detail, represented as key-value pair

### **Required Parameters**

#### **key**

- **type:** string
- **description:** Detail key

#### **value**

- **type:** string
- **description:** Detail value

### **Optional Parameters**

#### **critical**

- **type:** boolean

## **DiscoveredEntityApiDTO**

### **Description**

Model to describe a discovered entity in the system

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **environmentType**

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

### **discoveredBy**

- **description:** Target that discovered the entity
- **TargetApiDTO**

### **vendorIds**

- **type:** object
- **description:** The mapping of target identifier to vendor-provided identity of this entity on the remote target
- **additionalProperties:** {'type': 'string'}

## **EntityActionsApiDTO**

### **Description**

Basic informations of an entity + actions

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **realtimeMarketReference**

- **description:** In case of a Plan Market Entity, this contains the original entity from the Real Time Market
- **BaseApiDTO**

#### **actions**

- **type:** array
- **description:** List of actions for this entity
- – **ActionApiDTO**

## **EntityAspect**

### **Description**

Parent class for entity aspects

### **Required Parameters**

#### **type**

- **type:** string



## EntityCountRequestApiDTO

### Description

A query used to fetch entity counts.

### Optional Parameters

#### groupBy

- **type:** array
- **description:** Criteria to group the results by.
- **readOnly:** True
- – **type:** string
  - **enum:** ['ENTITY\_TYPE', 'ENVIRONMENT\_TYPE']

## EntityDetailsApiDTO

### Description

Model to describe an entity with all its details

### Required Parameters

#### uuid

- **type:** integer
- **format:** int64
- **description:** The internal UUID of the entity

### Optional Parameters

#### details

- **type:** array
- **description:** A list of details, represented as key-value
- – [DetailDataApiDTO](#)

## EntityMetadataRequestApiDTO

### Description

A query used to fetch entity metadata.

### Required Parameters

#### entityType

- **type:** string
- **description:** Entity type to fetch metadata for.
- **readOnly:** True
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray',

'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

## EntityPriceDTO

### Description

Context-specific price. For example the base rate of a template in the context of certain business unit.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### price

- **type:** number
- **format:** float
- **description:** Price in the context of a related entity

## EntityQueryApiDTO

### Description

A query used to search topology entities.

### Required Parameters

#### select

- **description:** Information to be fetched for each entity.
- **readOnly:** True
- **SelectEntityApiDTO**

### Optional Parameters

#### where

- **description:** Conditions that each retrieved entity must satisfy.
- **readOnly:** True
- **WhereApiDTO**

#### pagination

- **description:** Pagination information.

- **readOnly:** True
- **PaginationApiDTO**

## EntityStatsApiDTO

### Description

Basic informations of an entity + statistics

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

#### realtimeMarketReference

- **description:** In case of a Plan Market Entity, this contains the original entity from the Real Time Market
- **BaseApiDTO**

#### stats

- **type:** array
- **description:** Collected monitoring statistics
- – **StatSnapshotApiDTO**

## EntityUptimeApiDTO

### Description

Model to describe the uptime of an entity

### Optional Parameters

#### creationTimestamp

- **type:** integer
- **format:** int64
- **description:** The creation timestamp of the entity

#### uptimeDurationInMilliseconds

- **type:** integer
- **format:** int64

- **description:** Uptime duration in milliseconds for the total analyzed duration

#### **totalDurationInMilliseconds**

- **type:** integer
- **format:** int64
- **description:** Total analyzed duration in milliseconds for which uptime is calculated

#### **uptimePercentage**

- **type:** number
- **format:** double
- **description:** The uptime of the entity as a percentage

## **EnumFieldValueApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [FieldValueApiDTO](#)

### **Required Parameters**

#### **value**

- **type:** string
- **description:** The string representation of the value.
- **readOnly:** True

## **ExecutorInfoApiDTO**

### **Description**

Model to describe the entity that triggered action execution. such as a user or a schedule.

### **Required Parameters**

#### **type**

- **type:** string
- **description:** The type of the executor. such as a user or a schedule.
- **enum:** ['USER', 'SCHEDULE']

### **Optional Parameters**

#### **scheduleInfo**

- **description:** Information about the schedule that triggered this action. Omitted if the action was not triggered by a schedule.
- **[ExecutorScheduleApiDTO](#)**

#### **userInfo**

- **description:** Information about the user who triggered this action. Omitted if the action was not triggered by a user.
- **[ExecutorUserApiDTO](#)**

## ExecutorScheduleApiDTO

### Description

Model to describe the Schedule that executed the action.

### Required Parameters

#### scheduleName

- **type:** string
- **description:** Name of the schedule

## ExecutorUserApiDTO

### Description

Model to describe the user who triggered this action.

### Required Parameters

#### userName

- **type:** string
- **description:** Name of the user

## FailedResourceApiDTO

### Description

A description of the resource that caused an entity to be unplaced

### Required Parameters

#### commodity

- **description:** The commodity that caused the entity to be unplaced
- **BaseCommodityApiDTO**

#### requestedAmount

- **type:** number
- **format:** double
- **description:** The requested amount of the commodity that caused the entity to be unplaced

### Optional Parameters

#### maxAvailable

- **type:** number
- **format:** double
- **description:** The maximum available amount of the requested commodity. Populated only if there are one or more entities selling the requested commodity.

## FieldApiDTO

### Description

A field of an entity or group.

### Optional Parameters

#### fieldType

- **type:** string
- **description:** The kind of the field.
- **enum:** ['PRIMITIVE', 'COMMODITY', 'RELATED\_ENTITY', 'RELATED\_GROUP', 'RELATED\_ACTION', 'AGGREGATE\_COMMODITY', 'MEMBER']

## FieldValueApiDTO

### Description

A field of an entity or group with its value.

### Required Parameters

#### field

- **description:** Field to compare.
- **readOnly:** True
- **[FieldApiDTO](#)**

#### valueType

- **type:** string
- **description:** Type of compared values.
- **readOnly:** True
- **enum:** ['NUMBER', 'TEXT', 'ENUM', 'INTEGER', 'BOOLEAN', 'MULTI\_TEXT']

## FieldValueTypeApiDTO

### Description

A field of an entity or group associated with the type of its values.

### Required Parameters

#### field

- **description:** Field for which information is presented.
- **readOnly:** True
- **[FieldApiDTO](#)**

#### valueType

- **type:** string
- **description:** Type of values of this field.
- **readOnly:** True
- **enum:** ['NUMBER', 'TEXT', 'ENUM', 'INTEGER', 'BOOLEAN', 'MULTI\_TEXT']

## Optional Parameters

### allowableValues

- **type:** array
- **description:** Allowable string values for this field.
- **readOnly:** True
- – **type:** string

## FilterApiDTO

### Description

Model used to filter Groups or Entities, every filter is built using a filterType + expType + expVal

### Required Parameters

#### caseSensitive

- **type:** boolean
- **description:** Whether the regex is case sensitive

#### expType

- **type:** string
- **description:** Type of comparator used while building a filter, the allowable values are 'EQ', 'NEQ', 'RXEQ', 'RXNEQ' for strings, 'EQ', 'NEQ', 'GT', 'LT', 'GTE', 'LTE' for numbers
- **enum:** ['EQ', 'NEQ', 'GT', 'LT', 'GTE', 'LTE', 'RXEQ', 'RXNEQ']

#### expVal

- **type:** string
- **description:** Regex to be searched for

#### filterType

- **type:** string
- **description:** Internal name for criterion such as pmsByName, pmsByDC, storageByPMCluster

## Optional Parameters

### entityType

- **type:** string
- **description:** The entity type which this filter should be applied on

### singleLine

- **type:** boolean
- **description:** Whether the regex dot matches all characters including line break characters. Defaults to false.

## GroupApiDTO

### Description

Model to describe a Group: a collection of Entities. Subclasses include BillingFamilyApiDTO and ResourceGroupApiDTO.

## Required Parameters

### groupClassName

- **type:** string

### groupType

- **type:** string
- **description:** The type of service entities comprising the group

### isStatic

- **type:** boolean
- **description:** True if group is static

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### membersCount

- **type:** integer
- **format:** int32
- **description:** Number of members of the Group

### entitiesCount

- **type:** integer
- **format:** int32
- **description:** Number of entities of the Group

### costPrice

- **type:** number
- **format:** float
- **description:** Cost of the Group per Hour: sum of the costs of the member entities

### severity

- **type:** string
- **description:** Calculated using the highest severity of the member entities
- **enum:** ['UNKNOWN', 'NORMAL', 'MINOR', 'MAJOR', 'CRITICAL']

### state

- **type:** string
- **description:** Calculated using the state of the member entities
- **enum:** ['UNKNOWN', 'ACTIVE']

### environmentType

- **type:** string



- **description:** Environment type
- **enum:** ['HYBRID', 'CLOUD', 'ONPREM', 'UNKNOWN']

#### scope

- **type:** array
- **description:** Scope within which the criteria have to be applied to the list of specific entity type
- – **type:** string

#### logicalOperator

- **type:** string
- **description:** Logical operator to be applied across all the criteria used to create dynamic group.It can take values 'AND', 'OR', 'XOR'

#### criteriaList

- **type:** array
- **description:** Criteria list of elements, expression type, expression value, seType - needed to create a dynamic group
- – **FilterApiDTO**

#### memberUuidList

- **type:** array
- **description:** Uuid list for members of the group - if group is static
- – **type:** string

#### realtimeMarketReference

- **description:** In case of a Plan Market Group, this contains the original group from the Real Time Market
- **BaseApiDTO**

#### stats

- **type:** array
- **description:** Collected monitoring statistics
- – **StatSnapshotApiDTO**

#### temporary

- **type:** boolean
- **description:** The Group is only valid for a limited period of time, it will be removed automatically

#### activeEntitiesCount

- **type:** integer
- **format:** int32
- **description:** The active entities count of a group

#### cloudType

- **type:** string
- **description:** Type of the Cloud if environmentType = CLOUD.
- **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

#### source

- **description:** The Source of the Group
- **TargetApiDTO**

### **memberTypes**

- **type:** array
- **description:** The types for immediate members of the group.
- – **type:** string

### **entityTypes**

- **type:** array
- **description:** The types of entities contained in the group. This includes types of entities in nested levels of the group if the group is nested.
- – **type:** string

### **aspects**

- **type:** object
- **description:** Additional info about the Group categorized as Aspects
- **additionalProperties:** {'\$ref': '#/definitions/EntityAspect'}

### **vendorIds**

- **type:** object
- **description:** The mapping of target identifier to vendor-provided identity of this group, if the group is discovered
- **additionalProperties:** {'type': 'string'}

### **groupOrigin**

- **type:** string
- **description:** Origin of entities and groups
- **enum:** ['DISCOVERED', 'USER', 'SYSTEM']

## **GroupCountRequestApiDTO**

### **Description**

A query used to fetch group counts.

### **Optional Parameters**

#### **groupBy**

- **type:** array
- **description:** Criteria to group the results by.
- **readOnly:** True
- **uniqueItems:** True
- – **type:** string
- **enum:** ['GROUP\_TYPE', 'ORIGIN']

## **GroupMetadataRequestApiDTO**

### **Description**

A query used to fetch group metadata.

## Required Parameters

### groupType

- **type:** string
- **description:** Group type to fetch metadata for.
- **readOnly:** True
- **enum:** ['Group', 'Resource', 'Cluster', 'VMCluster', 'StorageCluster', 'BusinessAccountFolder', 'BillingFamily', 'NodePool']

## GroupQueryApiDTO

### Description

A query used to search groups.

### Required Parameters

#### select

- **description:** Information to be fetched for each group.
- **readOnly:** True
- **SelectGroupApiDTO**

### Optional Parameters

#### where

- **description:** Conditions that each retrieved group must satisfy.
- **readOnly:** True
- **WhereApiDTO**

#### pagination

- **description:** Pagination information.
- **readOnly:** True
- **PaginationApiDTO**

## HealthCategoryResponseDTO

### Description

A DTO to collect the aggregated health data for one health category.

### Required Parameters

#### categoryDisplayName

- **type:** string
- **description:** The reported health category name.

#### categoryHealthState

- **type:** string
- **description:** The health state for the reported category.
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

### healthCategory

- **type:** string
- **description:** The reported health category.
- **enum:** ['TARGET', 'ACTION', 'COMPONENT']

### Optional Parameters

#### responseItems

- **type:** array
- **description:** A list of individual health response items for this health category.
- – AggregatedHealthResponseDTO

## HttpProxyDTO

### Description

Model to describe http proxy settings, like username, password, port number and host

### Required Parameters

#### isProxyEnabled

- **type:** boolean
- **description:** Whether proxy is enabled or not

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### proxyHost

- **type:** string
- **description:** Proxy host

#### proxyPortNumber

- **type:** integer
- **format:** int32
- **description:** Proxy port number

#### userName

- **type:** string
- **description:** Proxy username

#### password

- **type:** string

- **description:** Proxy password

## HttpServletResponse

### Description

### Optional Parameters

#### status

- **type:** integer
- **format:** int32

#### headerNames

- **type:** array
- – **type:** string

#### bufferSize

- **type:** integer
- **format:** int32

#### contentType

- **type:** string

#### outputStream

- [ServletOutputStream](#)

#### locale

- [Locale](#)

#### writer

- [PrintWriter](#)

#### characterEncoding

- **type:** string

#### committed

- **type:** boolean

## IEntityDefinitionData

### Description

Parent class for topology data definition

### Required Parameters

#### definitionType

- **type:** string
- **enum:** ['MANUAL', 'AUTOMATED']

## IManualConnectionsData

### Description

Parent class for manual connections.

### Required Parameters

#### connectionType

- **type:** string
- **enum:** ['DYNAMIC', 'STATIC', 'GROUP']

## IncludedCouponsApiDTO

### Description

Included RI/Coupon and related information in the scenario

### Optional Parameters

#### includedCouponOidsList

- **type:** array
- **description:** User selected list of RIs to be included in plans
- – **type:** integer
- **format:** int64

#### iswhiteList

- **type:** boolean

## InclusionConditionApiDTO

### Description

### Inherits From

This DTO inherits from [ConditionApiDTO](#)

### Required Parameters

#### value

- **type:** array
- **description:** A set of strings. The value must belong to one of those.
- **readOnly:** True
- – **type:** string

## InputFieldApiDTO

### Description

Basic information of an entity

## Required Parameters

### name

- **type:** string
- **description:** Name of the field, used for field identification.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### value

- **type:** string
- **description:** Field value. Used if field holds primitive value (String, number or boolean)

### defaultValue

- **type:** string
- **description:** Default value of the field

### isMandatory

- **type:** boolean
- **description:** Whether the field is mandatory. Valid targets must have all the mandatory fields set.

### isSecret

- **type:** boolean
- **description:** Whether the field is secret. This means, that field value is stored in an encrypted value and not shown in any logs.

### isMultiline

- **type:** boolean
- **description:** Whether the field can have multiline values. Can be only true when the valueType field is set to STRING.

### isTargetDisplayName

- **type:** boolean
- **description:** Whether the field is used as a target display name.

### valueType

- **type:** string
- **description:** Type of the value this field holds
- **enum:** ['STRING', 'BOOLEAN', 'NUMERIC', 'INTEGER', 'LIST', 'OBJECT', 'GROUP\_SCOPE']

### specificValueType

- **type:** string
- **description:** A specific Type of the value that can't be described in valueType

### **description**

- **type:** string
- **description:** Additional information about what the input to the field should be

### **verificationRegex**

- **type:** string
- **description:** The regex pattern that needs to be satisfied for the input field text

### **groupProperties**

- **type:** array
- **description:** Group scope structure, filled if this field represents group scope value
- – **type:** array
  - - **type:** string

### **allowedValues**

- **type:** array
- **description:** Potential values for this field. If nonempty, value must be one of the choices in the list.
- – **type:** string

### **dependencyKey**

- **type:** string
- **description:** A key of a dependency account value field. If this value is not set, field is declared without any dependencies. If an annotated account value field has a dependency it means that it's presence is logically driven by the dependency field. This field is expected to be available ONLY when field specified by {@code dependencyKey} has value {@code dependencyValue}. If this field is not available because of unmet dependency, this field will not be validated and stored in target configuration data. If a dependency for this field is unmet value of this field will be null

### **dependencyValue**

- **type:** string
- **example:** one|two|four
- **description:** Value of a field of account values referenced by {@code dependencyKey} to enable (activate) the annotated field. Value is specified using a regular expression. Specifying just a string will also work correctly. If a referenced account value field is a boolean field, only "true" or "false" are supported as values. If a referenced account value is an enumeration ({@code allowedValues} is specified) this value is only restricted to contain a subset of the allowed values united using a pipe {@code one|two|four}.

## **InstanceTypeFilterApiDTO**

### **Description**

Model to describe instanceType filter information for the action filtering.

### **Optional Parameters**

#### **currentInstanceTypeFilter**

- **description:** Filter the actions by matching current instanceType and given the instanceType Uuid list.
- **InstanceTypeIdsApiDTO**

#### **newInstanceTypeFilter**

- **description:** Filter the actions by matching new instanceType and given the instanceType Uuid list.



- [InstanceTypeIdsApiDTO](#)

## InstanceTypeIdsApiDTO

### Description

Model to describe instanceType Uuid list and operator for the action filtering.

### Optional Parameters

#### instanceTypeIds

- **type:** array
- **description:** InstanceType Uuid list for the action filter.
- – **type:** string

#### operator

- **type:** string
- **description:** Operator for instanceType Uuid filter.the allowable values are 'EQ' and 'NEQ'
- **enum:** ['EQ', 'NEQ']

## IntegerConditionApiDTO

### Description

### Inherits From

This DTO inherits from [ConditionApiDTO](#)

### Required Parameters

#### value

- **type:** integer
- **format:** int64
- **description:** Value to compare against.
- **readOnly:** True

## IntegerFieldValueApiDTO

### Description

### Inherits From

This DTO inherits from [FieldValueApiDTO](#)

### Required Parameters

#### value

- **type:** integer
- **format:** int64
- **description:** The value.

- **readOnly:** True

## Optional Parameters

### units

- **type:** string
- **description:** Units of measurement.
- **readOnly:** True

## LicenseApiDTO

### Description

Model to describe properties of a license

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### licenseOwner

- **type:** string
- **description:** The owner of current license

#### email

- **type:** string
- **description:** The email of current license

#### customerId

- **type:** string
- **description:** The customer id

#### expirationDate

- **type:** string
- **description:** The expiration date of current license

#### features

- **type:** array
- **description:** List of features contained in current license
- **uniqueItems:** True
- – **type:** string

#### numLicensedEntities

- **type:** integer
- **format:** int32

- **description:** The number of licensed sockets

#### **numInUseEntities**

- **type:** integer
- **format:** int32
- **description:** The number of sockets in use

#### **edition**

- **type:** string
- **description:** The edition of the current license

#### **filename**

- **type:** string
- **description:** The name of the uploaded file

#### **countedEntity**

- **type:** string
- **description:** Indicates which entity type is counted with respect to licensing
- **enum:** ['VM', 'SOCKET']

#### **maxReportEditorsCount**

- **type:** integer
- **format:** int32
- **description:** Maximum number of report editors

#### **expirationDateTime**

- **type:** string
- **format:** date-time

#### **errorReasons**

- **type:** array
- **description:** Indicate why the license is invalid
- **uniqueItems:** True
- – **type:** string
  - **enum:** ['DUPLICATE\_LICENSE', 'EXPIRED', 'INVALID\_EMAIL', 'INVALID\_CONTENT\_TYPE', 'INVALID\_LICENSE\_KEY', 'INVALID\_LICENSE\_TYPE\_CWOM\_ONLY', 'INVALID\_LICENSE\_TYPE\_FREEMIUM\_NOT\_ALLOWED', 'INVALID\_FEATURE\_SET', 'INCOMPATIBLE']

#### **isValid**

- **type:** boolean
- **description:** Indicate if current license is valid or not
- **readOnly:** True

#### **isExpired**

- **type:** boolean
- **description:** Indicate if current license is expired
- **readOnly:** True

## Link

### Description

### Optional Parameters

#### rel

- **type:** string
- **xml:** {'attribute': True}

#### href

- **type:** string
- **xml:** {'attribute': True}

## LoadChangesApiDTO

### Description

Workload utilization changes in the scenario

### Optional Parameters

#### baselineDate

- **type:** string
- **description:** Baseline date

#### peakBaselineList

- **type:** array
- **description:** List of peak baseline changes
- – [PeakBaselineApiDTO](#)

#### utilizationList

- **type:** array
- **description:** List of utilization changes
- – [UtilizationApiDTO](#)

#### maxUtilizationList

- **type:** array
- **description:** List of max utilization changes
- – [MaxUtilizationApiDTO](#)

#### overlayStatsList

- **type:** array
- **description:** List of stat changes to override the plan commodities values with the provided input
- – [EntityStatsApiDTO](#)

## Locale

### Description

## Optional Parameters

### language

- **type:** string

### script

- **type:** string

### country

- **type:** string

### variant

- **type:** string

### extensionKeys

- **type:** array
- **uniqueItems:** True
- – **type:** string

### unicodeLocaleAttributes

- **type:** array
- **uniqueItems:** True
- – **type:** string

### unicodeLocaleKeys

- **type:** array
- **uniqueItems:** True
- – **type:** string

### iso3Language

- **type:** string

### iso3Country

- **type:** string

### displayLanguage

- **type:** string

### displayScript

- **type:** string

### displayCountry

- **type:** string

### displayVariant

- **type:** string

### displayName

- **type:** string

## LogEntryApiDTO

### Description

Basic information of an entity

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### logActionTime

- **type:** integer
- **format:** int64

### targetSE

- **type:** string

### targetUuid

- **type:** string

### currentSE

- **type:** string

### currentSEUuid

- **type:** string

### newSE

- **type:** string

### newSEUuid

- **type:** string

### category

- **type:** string

### subCategory

- **type:** string

### description

- **type:** string

### shortDescription

- **type:** string

### state

- **type:** string

### severity

- **type:** string

### count

- **type:** integer
- **format:** int64

**total**

- **type:** integer
- **format:** int64

**user**

- **type:** string

**type**

- **type:** string

**importance**

- **type:** number
- **format:** float

**reasonCommodities**

- **type:** array
- **description:** The distinct set of commodities that were the reason for the action. Not all actions are driven by commodities. So its possible that this can be an empty list.
- **uniqueItems:** True
- – **type:** string

**reasonCommodityRelationType**

- **type:** string
- **description:** Relation type (bought/sold) of the reason commodity
- **enum:** ['sold', 'bought']

**logDBId**

- **type:** integer
- **format:** int64

## LoggingApiDTO

### Description

Model to describe the component logging level

### Optional Parameters

**componentLoggingLevel**

- **type:** object
- **description:** A map from component name to logging level, component could be Discovery, Monitoring, Presentation, Abstraction, Analysis, Extension; logging levels could be INFO, WARN, DEBUG, TRACE
- **additionalProperties:** {'type': 'string', 'enum': ['FATAL', 'ERROR', 'WARN', 'INFO', 'DEBUG', 'TRACE', 'UNKNOWN']}

## ManualDynamicConnections

### Description

## Inherits From

This DTO inherits from [IManualConnectionsData](#)

## Optional Parameters

### dynamicConnectionCriteria

- **type:** array
- **description:** Connection criteria to define dynamically connected entities.
- – [FilterApiDTO](#)

## ManualEntityDefinitionData

### Description

### Inherits From

This DTO inherits from [IEntityDefinitionData](#)

### Required Parameters

#### manualConnectionData

- **type:** object
- **description:** Entity connections set manually
- **additionalProperties:** {'\$ref': '#/definitions/IManualConnectionsData'}

## ManualGroupConnections

### Description

### Inherits From

This DTO inherits from [IManualConnectionsData](#)

### Required Parameters

#### connectedGroup

- **description:** Group for a connection.
- [BaseGroupApiDTO](#)

## ManualStaticConnections

### Description

### Inherits From

This DTO inherits from [IManualConnectionsData](#)



## Optional Parameters

### staticConnections

- **type:** array
- **description:** Oids to be statically connected to the created entity
- – **type:** string

## MarketApiDTO

### Description

Model to describe a Market

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### state

- **type:** string
- **description:** State
- **enum:** ['CREATED', 'READY\_TO\_START', 'RUNNING', 'COPYING', 'SUCCEEDED', 'STOPPING', 'STOPPED', 'DELETING']

#### stateProgress

- **type:** integer
- **format:** int32
- **description:** State progress indicated by a percentage, used in RUNNING state

#### scenario

- **description:** Scenario used to create the Plan, only for Plan Market
- **ScenarioApiDTO**

#### saved

- **type:** boolean
- **description:** Shows if a Plan Market is saved

#### unplacedEntities

- **type:** boolean
- **description:** Unplaced entities, shows if there are entities that could not be placed in the Plan Market

#### runDate

- **type:** string
- **description:** Shows when a Plan Market started running

### runCompleteDate

- **type:** string
- **description:** Shows when a Plan Market was completed

### violations

- **type:** array
- **description:** Violations generated in a Plan Market
- – [ServiceEntityApiDTO](#)

### relatedPlanMarkets

- **type:** array
- **description:** Related Plan Markets generated by particular Scenarios
- – [MarketApiDTO](#)

### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['HYBRID', 'CLOUD', 'ONPREM']

### planDestination

- **description:** Plan destination where the plan results would be stored, only for Plan Market.
- [BaseApiDTO](#)

## MasterImageEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### displayName

- **type:** string
- **description:** DisplayName of the masterImage used to create new VMs

#### numVcpus

- **type:** integer
- **format:** int32
- **description:** The number of vCPUs allocated to each VM generated from the master image

#### mem

- **type:** number
- **format:** float
- **description:** The amount of Mem allocated to each VM generated from the master image

#### storage

- **type:** number
- **format:** float

- **description:** The amount of Storage allocated to each VM generated from the master image

## MaxUtilizationApiDTO

### Description

Model to describe a max utilization change in the scenario

### Optional Parameters

#### projectionDay

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### target

- **description:** Target of the change, entity or group. If empty, assumes global scope
- **BaseApiDTO**

#### maxPercentage

- **type:** integer
- **format:** int32
- **description:** Max utilization threshold percentage change

#### selectedEntityType

- **type:** string
- **description:** Type of entity to select in global scope configuration

#### globalScope

- **type:** boolean

## MemberFieldApiDTO

### Description

### Inherits From

This DTO inherits from [FieldApiDTO](#)

### Required Parameters

#### property

- **type:** string
- **description:** The property of interest.
- **readOnly:** True
- **enum:** ['COUNT']

### Optional Parameters

#### entityType

- **type:** string

- **description:** Restricts the property to entities of this type.
- **readOnly:** True
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

#### direct

- **type:** boolean
- **description:** Whether this field refers to direct or indirect members (default = direct).
- **readOnly:** True

## MigrateObjectApiDTO

### Description

Model to describe a replace change in the scenario

### Optional Parameters

#### projectionDay

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### source

- **description:** Source of the migration, entity or group - [DEPRECATED - Use: sources]
- **BaseApiDTO**

#### destination

- **description:** Destination of the migration - [DEPRECATED - Use: destinations]
- **BaseApiDTO**

#### sources

- **type:** array
- **description:** Sources of the migration, entities or groups.
- – **BaseApiDTO**

#### destinations

- **type:** array
- **description:** Destinations of the migration, entities or groups.
- – **BaseApiDTO**

#### destinationEntityType

- **type:** string

- **description:** Entity type of migration destination, default is VirtualMachine. Case 1, migration source VirtualMachine: destination entity must be VirtualMachine. Case 2, migration source Database Server: destination entity can be VirtualMachine or DatabaseServer.
- **enum:** ['VirtualMachine', 'DatabaseServer']

#### **removeNonMigratingWorkloads**

- **type:** boolean
- **description:** If true, remove the existing Virtual Machines, Databases and Database Servers from the scope that are not migrating, so that only the newly migrated entities are included in the plan results.

## **MultiTextFieldValueApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [FieldValueApiDTO](#)

### **Required Parameters**

#### **value**

- **type:** array
- **description:** The values associated with the field.
- **readOnly:** True
- **– type:** string

## **NameValueInputDTO**

### **Description**

Deployment Profile provider parameters

### **Optional Parameters**

#### **name**

- **type:** string
- **description:** Parameter type. For vCenter, this can be a name or id. For all other target types, this must be an id.
- **enum:** ['name', 'id']

#### **value**

- **type:** string

## **NumberConditionApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [ConditionApiDTO](#)

## Required Parameters

### value

- **type:** number
- **format:** double
- **description:** Value to compare against.
- **readOnly:** True

## NumberFieldValueApiDTO

### Description

### Inherits From

This DTO inherits from [FieldValueApiDTO](#)

## Required Parameters

### value

- **type:** number
- **format:** double
- **description:** The value.
- **readOnly:** True

## Optional Parameters

### units

- **type:** string
- **description:** Units of measurement.
- **readOnly:** True

## OAuthDataApiDTO

### Description

## Required Parameters

### authorizationServerUrl

- **type:** string
- **description:** The URL of the authorization server.

### clientId

- **type:** string
- **description:** The client id used for oAuth authorization.

### clientSecret

- **type:** string
- **description:** The client secret used for oAuth authorization.

### grantType

- **type:** string
- **description:** The oAuth grant type.
- **enum:** ['CLIENT\_CREDENTIALS']

### Optional Parameters

#### scope

- **type:** string
- **description:** The oAuth scope.

## OnDemandCostFilterApiDTO

### Description

Model to describe on-demand cost range for action filtering.

### Optional Parameters

#### currentOnDemandCostRange

- **description:** Filter the actions by the range for current on-demand cost.
- **RangeInputApiDTO**

#### newOnDemandCostRange

- **description:** Filter the actions by the range for destination on-demand cost. Applicable only for scale actions.
- **RangeInputApiDTO**

## OnPremResizeActionDetailsApiDTO

### Description

### Inherits From

This DTO inherits from [ActionDetailsApiDTO](#)

### Optional Parameters

#### vcpuBefore

- **type:** integer
- **format:** int32
- **description:** vCPUs before the change

#### vcpuAfter

- **type:** integer
- **format:** int32
- **description:** vCPUs after the change

#### socketsBefore

- **type:** integer
- **format:** int32

- **description:** sockets before the change

#### **socketsAfter**

- **type:** integer
- **format:** int32
- **description:** sockets after the change

#### **coresPerSocketBefore**

- **type:** integer
- **format:** int32
- **description:** cores per socket before the change

#### **coresPerSocketAfter**

- **type:** integer
- **format:** int32
- **description:** cores per socket after the change

#### **nonDisruptiveMaxPU**

- **type:** number
- **format:** double
- **description:** Non-disruptive Processing Units max resize constraint

#### **nonDisruptiveMinPU**

- **type:** number
- **format:** double
- **description:** Non-disruptive Processing Units min resize constraint

#### **subType**

- **type:** string

## **OrderByApiDTO**

### **Description**

A field to order results by.

### **Required Parameters**

#### **field**

- **description:** The field to order by.
- **readOnly:** True
- **FieldApiDTO**

### **Optional Parameters**

#### **ascending**

- **type:** boolean
- **description:** Whether the order is ascending (defaults to false).
- **readOnly:** True



## PMDiskAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### diskCapacity

- **type:** integer
- **format:** int64
- **description:** Disk Capacity

#### diskRole

- **type:** string
- **description:** Disk Role
- **enum:** ['CAPACITY', 'CACHE']

## PMDiskGroupAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### disks

- **type:** array
- **description:** Disks
- – [PMDiskAspectApiDTO](#)

## PMEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### diskGroups

- **type:** array
- **description:** Disk Groups
- – [PMDiskGroupAspectApiDTO](#)

**dedicatedFailoverHost**

- **type:** boolean
- **description:** If true, this host is a failover host.

**cpuModel**

- **type:** string
- **example:** Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz
- **description:** The CPU model of the host

**connectedNetworks**

- **type:** array
- **example:** Broadcom BCM5708C NetXtreme II GigE
- **description:** Connected Networks
- – **type:** string

**automationLevel**

- **type:** string
- **example:** Fully Automated
- **description:** Vendor provided automation level
- **enum:** ['FULLY\_AUTOMATED', 'PARTIALLY\_AUTOMATED', 'NOT\_AUTOMATED', 'DISABLED']

**migrationLevel**

- **type:** string
- **example:** Conservative
- **description:** Vendor provided migration level

**machineTypeAndModel**

- **type:** string
- **example:** 9040-MR9
- **description:** Machine Type and Model

**serialNumber**

- **type:** string
- **example:** 130330X
- **description:** Serial Number

**totalInstalledPhysicalProcessorUnits**

- **type:** integer
- **format:** int32
- **example:** 80
- **description:** Total Installed Physical Processor Units

**supportedProcessorCompatibilityModes**

- **type:** string
- **example:** POWER7, POWER8, POWER9
- **description:** Supported Processor Compatibility Modes

**latestSupportedProcessorGeneration**

- **type:** string
- **example:** 9

- **description:** Latest Supported Processor Generation

#### **totalInstalledPhysicalMemory**

- **type:** integer
- **format:** int32
- **example:** 4096
- **description:** Total Installed Physical Memory

#### **hypervisorUsedMemory**

- **type:** integer
- **format:** int32
- **example:** 4096
- **description:** Hypervisor Used Memory

#### **memoryAssignedToVMs**

- **type:** integer
- **format:** int32
- **example:** 4096
- **description:** Memory Assigned To VMs

#### **logicalMemoryBlockSize**

- **type:** integer
- **format:** int32
- **example:** 256
- **description:** Logical Memory Block Size

#### **processorPools**

- **type:** array
- – **type:** string

## **PaginationApiDTO**

### **Description**

Pagination parameters.

### **Optional Parameters**

#### **orderBy**

- **type:** array
- **description:** Order by these fields.
- **readOnly:** True
- – **OrderByApiDTO**

#### **cursor**

- **type:** string
- **description:** Cursor for the next batch.
- **readOnly:** True

#### **limit**

- **type:** integer

- **format:** int32
- **description:** Maximum number of records to fetch.
- **readOnly:** True

## ParkableEntityApiDTO

### Description

Parkable Entity

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

#### accountOID

- **type:** integer
- **format:** int64
- **example:** 74613608411667
- **description:** Unique ID of the account

#### accountName

- **type:** string
- **example:** dev-account
- **description:** Display name of the account

#### provider

- **type:** string
- **example:** AWS
- **description:** Name of the cloud provider
- **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

#### state

- **type:** string
- **example:** STOPPED
- **description:** Last known state of the entity
- **enum:** ['RUNNING', 'STOPPED', 'SUSPENDED', 'MAINTENANCE', 'FAILOVER', 'UNKNOWN', 'STARTING', 'STOPPING']

**parkable**

- **type:** boolean
- **example:** True
- **description:** Signifies whether the entity can be parked

**entityType**

- **type:** string
- **example:** COMPUTE
- **description:** Type of the entity
- **enum:** ['VirtualMachine', 'DatabaseServer']

**regionOID**

- **type:** integer
- **format:** int64
- **example:** 74613608411668
- **description:** Unique ID of the region, the entity belongs to

**regionName**

- **type:** string
- **example:** us-east-1
- **description:** Name of the region

**instanceType**

- **type:** string
- **example:** t3.micro
- **description:** Type of the instance

**cost**

- **type:** number
- **format:** double
- **example:** 10.24
- **description:** On demand hourly cost of the entity

**tags**

- **type:** array
- **description:** List of tags attached to the entity
- – [TagApiDTO](#)

**schedules**

- **type:** array
- **description:** List of schedules attached to the entity
- – [ParkableEntityScheduleApiDTO](#)

**groups**

- **type:** array
- **description:** List of the Groups attached to the entity
- – [ParkableEntityGroupApiDTO](#)

**guestOSName**

- **type:** string

- **example:** Linux
- **description:** Guest OS Name of the entity

## ParkableEntityEnforcementPauseApiDTO

### Description

represents the partial failure per entity for enforcement pause on list of parkable entities

### Optional Parameters

#### error

- **type:** string
- **example:** entityId not found
- **description:** error indicating the reason for failure

#### entityUUID

- **type:** string

## ParkableEntityEnforcementPauseInputDTO

### Description

represents the enforcement pause on list of parkable entities either until a given time or in duration of minutes

### Optional Parameters

#### until

- **type:** string
- **example:** 2022-11-29T01:02:03+00:00
- **description:** Time when the enforcement pause on parkable entities ends. Should be in ISO 8601 format

#### durationMinutes

- **type:** integer
- **format:** int32
- **example:** 60
- **description:** duration in minutes to keep enforcement pause on parkable entities

#### entityUuids

- **type:** array
- **description:** list of parkable entities to pause enforcement on
- – **type:** string

## ParkableEntityGroupApiDTO

### Description

represents groups which are attached to the parkable entity

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### groupType

- **type:** string
- **example:** Resource
- **description:** Type of the Group
- **enum:** ['RESOURCE']

## ParkableEntityInputDTO

### Description

## Optional Parameters

### entityTypes

- **type:** array
- – **type:** string
  - **enum:** ['VirtualMachine', 'DatabaseServer']

### status

- **type:** array
- – **type:** string
  - **enum:** ['RUNNING', 'STOPPED', 'SUSPENDED', 'MAINTENANCE', 'FAILOVER', 'UNKNOWN', 'STARTING', 'STOPPING']

### providers

- **type:** array
- **description:** Parking Cloud Providers
- – **type:** string
  - **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

### criteriaList

- **type:** array
- – **FilterApiDTO**

### logicalOperator

- **type:** string
- **description:** Logical operator to be applied across all the criteria. It can take values 'AND', 'OR'. If omitted, defaults to 'AND'
- **enum:** ['AND', 'OR']

## ParkableEntityScheduleApiDTO

### Description

Represents time span based schedule display name and uuid of the time span based schedule attached to the parkable entity

### Optional Parameters

#### displayName

- **type:** string
- **example:** Up 7-5 on weekdays
- **description:** Display name of the time span based schedule the parkable entities is attached to

#### uuid

- **type:** string
- **example:** 74613608411667
- **description:** UUID of the time span based schedule the parkable entities are attached to

#### pauseEnforcementUntil

- **type:** integer
- **format:** int64
- **example:** 1673273639
- **description:** Pause Enforcement time over the entity, in epoch time format

#### attachedBy

- **description:** Details of the user/policy responsible for attaching the schedule to entity
- [ParkingEntitySchedulerApiDTO](#)

## ParkableEntityUUIDSetDTO

### Description

list of parkable entity uuids

### Required Parameters

#### entityUuids

- **type:** array
- **description:** list of parkable entity uuids
- – **type:** string

## ParkingEntitySchedulerApiDTO

### Description

Represents the details of the user/policy responsible for attaching the schedule to parkable entity

### Optional Parameters

#### uuid

- **type:** string



**displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**type**

- **type:** string
- **example:** POLICY
- **description:** Whether user or policy was responsible for attaching the schedule
- **enum:** ['USER', 'POLICY']

## ParkingItemApiDTO

### Description

### Inherits From

This DTO inherits from [ScheduleItemApiDTO](#)

### Required Parameters

**state**

- **type:** string
- **description:** expected state during the time span
- **enum:** ['IGNORE', 'ON', 'OFF']

## ParkingPolicyApiDTO

### Description

Model to describe a parking policy, used to assign timespan based schedules to entities based on global, cloud provider, account and resource group level filters

### Required Parameters

**level**

- **type:** string
- **description:** Specifies the level at which the policy is applicable.
- **enum:** ['GLOBAL', 'CLOUD\_PROVIDER', 'ACCOUNT', 'RESOURCE\_GROUP']

**timespanSchedule**

- **description:** timespan based schedule to be/attached to policy.
- **BaseApiDTO**

### Optional Parameters

**uuid**

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### scopes

- **type:** array
- **description:** List of scope object to which the parking policy is applicable, like details of account for account level policies.
- – **BaseApiDTO**

### criteriaList

- **type:** array
- **description:** List of filters to get the desired the entities to which the policy is applicable.
- – **FilterApiDTO**

### priority

- **type:** integer
- **format:** int32
- **description:** Priority is used to identify the applicable priority if there are multiple policies at same level

## PatchedTargetDataApiDTO

### Description

Model to describe a patched target data.

### Required Parameters

#### patchedFields

- **type:** array
- **description:** Patched target fields.
- – **PatchedTargetFieldApiDTO**

#### probeType

- **type:** string
- **description:** Probe type of the patched target.

## PatchedTargetFieldApiDTO

### Description

Model to describe a field in patched target data.

### Optional Parameters

#### fieldName

- **type:** string

- **description:** Patched field name.

#### **fieldValue**

- **type:** string
- **description:** Patched field value.

## **PeakBaselineApiDTO**

### **Description**

Model to describe a peak baseline change in the scenario

### **Optional Parameters**

#### **date**

- **type:** string
- **description:** Date of the utilization peak

#### **target**

- **description:** Target of the change
- **BaseApiDTO**

## **PlacementInfoDTO**

### **Description**

### **Optional Parameters**

#### **computeResources**

- **type:** array
- **description:** Compute resources that the reservation is placed in
- – **ResourceApiDTO**

#### **storageResources**

- **type:** array
- **description:** Storage resources that the reservation is placed in
- – **ResourceApiDTO**

#### **networkResources**

- **type:** array
- **description:** Network resources that the reservation is placed in
- – **ResourceApiDTO**

#### **failureInfos**

- **type:** array
- **description:** Details about why the reservation failed. This will only be populated if the reservation failed.
- – **ReservationFailureInfoDTO**

## invalidInfo

- **description:** Details about why the reservation is invalid. This will only be populated if the reservation is invalid.
- **ReservationInvalidInfoApiDTO**

# PlacementOptionApiDTO

## Description

Model to describe the property of provider options.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### constraintType

- **type:** string
- **description:** constraint type, if a commodity is AccessCommodity, then the class name of that commodity is a constraint type
- **enum:** ['ApplicationCommodity', 'ClusterCommodity', 'DataCenterCommodity', 'DatastoreCommodity etc']

### scope

- **description:** scope or placement policy
- **BaseApiDTO**

### target

- **description:** for placement policy, if it's an user placement policy is null.
- **TargetApiDTO**

### numPotentialEntities

- **type:** integer
- **format:** int32
- **description:** example: other VMs, which buys this constraint but not on this host

### key

- **type:** string
- **description:** commodity key, this information is needed when trying to retrieve the intersaction of providers or consumers

### currentEntities

- **type:** array
- **description:** for consumers only. example: VMs, which are currently on this host per constraint they are buying

- – ServiceEntityApiDTO

## PlacementOptionApiInputDTO

### Description

Model to describe the property of provider options.

### Optional Parameters

#### constraintType

- **type:** string
- **description:** constraint type, if a commodity is AccessCommodity, then the class name of that commodity is a constraint type
- **enum:** ['ApplicationCommodity', 'ClusterCommodity', 'DataCenterCommodity', 'DatastoreCommodity etc']

#### key

- **type:** string
- **description:** commodity key, this information is needed when trying to retrieve the intersection of providers or consumers

## PlacementParametersDTO

### Description

### Required Parameters

#### count

- **type:** integer
- **format:** int32
- **description:** Number of Virtual Machines to Place

### Optional Parameters

#### templateID

- **type:** string
- **description:** Template Uuid used for the Placement

#### entityNames

- **type:** array
- **description:** Name of the Placed Virtual Machine
- – **type:** string

#### constraintIDs

- **type:** array
- **description:** List of uuids that will define the constraints that new VMs should be compliant to, allowed uuids are: Cluster, Datacenter, VirtualDataCenter, Network, Placement Policy. The possible providers are the intersection of all constraints passed.
- **uniqueItems:** True
- – **type:** string

## **geographicRedundancy**

- **type:** boolean
- **description:** Enable Geographic Redundancy

# **PlanDestinationApiDTO**

## **Description**

Model to describe a plan destination. A plan destination is an external destination where plan results can be uploaded.

## **Required Parameters**

### **businessUnit**

- **description:** Business account to plan destination.
- **BaseApiDTO**

### **exportDescription**

- **type:** string
- **description:** Plan export description message on this plan destination.

### **exportProgressPercentage**

- **type:** integer
- **format:** int32
- **description:** Plan export progress percentage on this plan destination.

### **exportState**

- **type:** string
- **description:** Plan export state on this plan destination.
- **enum:** ['NONE', 'REJECTED', 'IN\_PROGRESS', 'SUCCEEDED', 'FAILED']

### **hasExportedData**

- **type:** boolean
- **description:** Boolean, true if this plan destination has uploaded data.

### **market**

- **description:** Plan market currently uploaded to plan destination.
- **BaseApiDTO**

### **numErrors**

- **type:** integer
- **format:** int32
- **description:** Number of errors on the plan destination.

## **Optional Parameters**

### **uuid**

- **type:** string

### **displayName**

- **type:** string

- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

## **PolicyApiDTO**

### **Description**

Model to describe a Workload policy, used to manage the entities in the Environment. e.g. Restrict the number of VMs that can use a given datastore, Limit VM mobility by restricting a collection of VMs to a specified cluster or group of hosts, Expand VM mobility by merging a number of clusters, Keep workload running on hosts that have the required licenses

### **Required Parameters**

#### **type**

- **type:** string
- **description:** Placement Policy Type
- **enum:** ['AT\_MOST\_N', 'BIND\_TO\_GROUP', 'BIND\_TO\_COMPLEMENTARY\_GROUP', 'MUST\_RUN\_TOGETHER', 'AT\_MOST\_N\_BOUND', 'MERGE', 'BIND\_TO\_GROUP\_AND\_LICENSE', 'MUST\_NOT\_RUN\_TOGETHER', 'EXCLUSIVE\_BIND\_TO\_GROUP']

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **name**

- **type:** string
- **description:** Name of the policy

#### **enabled**

- **type:** boolean
- **description:** Shows if the policy is enabled

#### **capacity**

- **type:** integer
- **format:** int32
- **description:** Indicate the max workload entities at the same time per placement entites, used when type is AT\_MOST\_N\_BOUND

#### **commodityType**

- **type:** string
- **description:** Commodity type used for the policy

### mergeType

- **type:** string
- **description:** Type of Group that is merged, used only when type is MERGE
- **enum:** ['Cluster', 'StorageCluster', 'DataCenter', 'DesktopPool']

### consumerGroup

- **description:** Consumer Group, used when type is any but MERGE
- **BaseApiDTO**

### providerGroup

- **description:** Provider Group, used when type is any but MERGE
- **BaseApiDTO**

### mergeGroups

- **type:** array
- **description:** List of Groups to merge, used when type is MERGE
- – **BaseApiDTO**

### providerEntityType

- **type:** string
- **description:** The entity type for the provider. This property is only applicable when policy type is set to MUST\_RUN\_TOGETHER or MUST\_NOT\_RUN\_TOGETHER. When policy type is set to MUST\_RUN\_TOGETHER or MUST\_NOT\_RUN\_TOGETHER, this field should be set with type of provider where members of consumer group must run or must not run together on it depending on the policy type.
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

## PolicyApiInputDTO

### Description

Model to describe a Placement Policy

### Required Parameters

#### policyName

- **type:** string
- **description:** Name of the policy

#### type

- **type:** string
- **description:** Placement Policy Type



- **enum:** ['AT\_MOST\_N', 'BIND\_TO\_GROUP', 'BIND\_TO\_COMPLEMENTARY\_GROUP', 'MUST\_RUN\_TOGETHER', 'AT\_MOST\_N\_BOUNDED', 'MERGE', 'BIND\_TO\_GROUP\_AND\_LICENSE', 'MUST\_NOT\_RUN\_TOGETHER', 'EXCLUSIVE\_BIND\_TO\_GROUP']

## Optional Parameters

### sellerUuid

- **type:** string
- **description:** Uuid of the provider Group in a Placement Policy

### buyerUuid

- **type:** string
- **description:** Uuid of the consumer Group in a Placement Policy

### mergeUuids

- **type:** array
- **description:** Uuids of the Groups in a Merge Policy
- – **type:** string

### mergeType

- **type:** string
- **description:** Group Type in a Merge Policy
- **enum:** ['Cluster', 'StorageCluster', 'DataCenter', 'DesktopPool']

### capacity

- **type:** integer
- **format:** int32
- **description:** For an AT\_MOST\_N or AT\_MOST\_N\_BOUNDED policy, the number of consumers to allow on a provider entity

### enabled

- **type:** boolean
- **description:** Whether to enable the policy

### providerEntityType

- **type:** string
- **description:** The entity type for the provider. This property is only applicable when the policy type is set to MUST\_RUN\_TOGETHER or MUST\_NOT\_RUN\_TOGETHER. This field is set with the provider type (for example, PhysicalMachine) the policy should respect.
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

# PortChannelApiDTO

## Description

Model to describe a Port Channel

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### capacity

- **description:** Capacity values
- **StatValueApiDTO**

### values

- **description:** Contains Average, Min, Max, Total
- **StatValueApiDTO**

### units

- **type:** string
- **description:** Units, used for Commodities stats. E.G. \$/h

### ports

- **type:** array
- **description:** List of ports in the channel
- – **StatApiDTO**

### filters

- **type:** array
- **description:** describe the grouping options used to generate the output
- – **StatFilterApiDTO**

# PortsAspectApiDTO

## Description

## Inherits From

This DTO inherits from [EntityAspect](#)

## Optional Parameters

### portChannels

- **type:** array
- **description:** Port Channels
- – [PortChannelApiDTO](#)

### ports

- **type:** array
- **description:** Independent ports
- – [StatApiDTO](#)

## PriceAdjustmentDTO

### Description

### Optional Parameters

#### type

- **type:** string
- **enum:** ['DISCOUNT', 'INCREASE']

#### value

- **type:** number
- **format:** float

## PrimitiveFieldApiDTO

### Description

### Inherits From

This DTO inherits from [FieldApiDTO](#)

### Required Parameters

#### fieldName

- **type:** string
- **description:** The name of the primitive field.
- **readOnly:** True

## PrintWriter

### Description

This is an unspecified object.

## ProbeApiDTO

### Description

Model to describe a probe.

### Required Parameters

#### category

- **type:** string
- **description:** Probe category.

#### id

- **type:** integer
- **format:** int64
- **description:** Probe id.

#### type

- **type:** string
- **description:** Probe type.

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### error

- **type:** string
- **description:** Unexpected error while collecting probe information

## ProbePropertyApiDTO

### Description

Model to describe a probe property

### Required Parameters

#### name

- **type:** string
- **description:** Probe property name.

#### value

- **type:** string
- **description:** Probe property value.

## Optional Parameters

### probeId

- **type:** integer
- **format:** int64
- **description:** Probe id.

### targetId

- **type:** integer
- **format:** int64
- **description:** Target id.

## ProbePropertyNameValuePairApiDTO

### Description

Model to describe a name/value pair for probe properties.

### Required Parameters

#### name

- **type:** string
- **description:** Probe property name.

#### value

- **type:** string
- **description:** Probe property value.

## ProductCapabilityDTO

### Description

Model to describe Product Capability information.

### Optional Parameters

#### deploymentMode

- **type:** string
- **description:** Deployment mode, affects hiding / showing multiple UI views, options
- **enum:** ['HYBRID\_SAAS', 'SAAS', 'SERVER', 'SAAS\_TRIAL']

#### searchApiEnabled

- **type:** boolean
- **description:** Controls use of new search api

#### parkingEnabled

- **type:** boolean
- **description:** Controls hiding/showing parking related pages in UI

#### parkingPoliciesEnabled

- **type:** boolean
- **description:** [BETA]: Controls hiding/showing parking policies related features in UI

### **frontEndTelemetryEnabled**

- **type:** boolean
- **description:** Controls enabling/disabling front end telemetry in UI

### **jdbcHttpSessionEnabled**

- **type:** boolean
- **description:** Controls use of JDBC persistent HTTP Session

### **saasReportingEnabled**

- **type:** boolean
- **description:** Controls hiding/showing SaaS reports page in UI

### **trialExpirationTime**

- **type:** string
- **description:** [BETA]: When the user's trial expires in ISO 8601 Format. For SAAS\_TRIAL deploymentMode

### **reportingEnabled**

- **type:** boolean

## **ProductVersionDTO**

### **Description**

Contains version information about the running Turbonomic instance.

### **Optional Parameters**

#### **versionInfo**

- **type:** string
- **example:** "versionInfo": "Turbonomic Operations Manager  
8.0.1-SNAPSHOT (Build \"20200922092548000\") \"2020-09-22  
20:10:04\"\\n\\naction-orchestrator: 8.0.1-SNAPSHOT\\napi: 8.0.1-SNAPSHOT\\nauth:  
8.0.1-SNAPSHOT\\ncost: 8.0.1-SNAPSHOT\\nextractor: 8.0.1-SNAPSHOT\\ngroup:  
8.0.1-SNAPSHOT\\nhistory: 8.0.1-SNAPSHOT\\nmarket: 8.0.1-SNAPSHOT\\nmediation-  
actionsript: 8.0.1-SNAPSHOT\\nmediation-appdynamics: 8.0.1-SNAPSHOT\\nmediation-  
appinsights: 8.0.1-SNAPSHOT\\nmediation-aws: 8.0.1-SNAPSHOT\\nmediation-  
awsbilling: 8.0.1-SNAPSHOT\\nmediation-awscost: 8.0.1-SNAPSHOT\\nmediation-  
azure: 8.0.1-SNAPSHOT\\nmediation-azurecost: 8.0.1-SNAPSHOT\\nmediation-azureea:  
8.0.1-SNAPSHOT\\nmediation-azuresp: 8.0.1-SNAPSHOT\\nmediation-azurevolumes:  
8.0.1-SNAPSHOT\\nmediation-dynatrace: 8.0.1-SNAPSHOT\\nmediation-hpe3par:  
8.0.1-SNAPSHOT\\nmediation-hyperv: 8.0.1-SNAPSHOT\\nmediation-netapp: 8.0.1-  
SNAPSHOT\\nmediation-nutanix: 8.0.1-SNAPSHOT\\nmediation-oneview: 8.0.1-  
SNAPSHOT\\nmediation-pure: 8.0.1-SNAPSHOT\\nmediation-ucs: 8.0.1-SNAPSHOT\\nmediation-  
vcenter: 8.0.1-SNAPSHOT\\nmediation-vcenterbrowsing: 8.0.1-SNAPSHOT\\nmediation-  
vmax: 8.0.1-SNAPSHOT\\nmediation-vmm: 8.0.1-SNAPSHOT\\nplan-orchestrator: 8.0.1-  
SNAPSHOT\\nrepository: 8.0.1-SNAPSHOT\\ntopology-processor: 8.0.1-SNAPSHOT"
- **description:** A multi-line string describing the version information, including that of the various internal components.

#### **version**

- **type:** string
- **example:** 8.0.1-SNAPSHOT
- **description:** The Operations Manager product version, including major and minor versions

**build**

- **type:** string
- **example:** 20200922211058775
- **description:** The build number associated with this deployment.

**commit**

- **type:** string
- **example:** 1183dd2
- **description:** The commit associated with this deployment.

**branch**

- **type:** string
- **example:** 8.0.1
- **description:** The branch associated with this deployment.

**gitDescription**

- **type:** string
- **example:** 1183dd2bb8ebca3dbcaf708e0c323cab7b8e352
- **description:** Specific source control information associated with this deployment.

**updates**

- **type:** string
- **description:** Description of available updates--not yet implemented in XL.

**marketVersion**

- **type:** integer
- **format:** int32
- **example:** 2
- **description:** The version of the market

**apiVersion**

- **type:** string
- **example:** 3.1
- **description:** The version of the REST API.

## PurchaseProfileDTO

**Description**

Model to describe a set of preferences for purchasing RIs

**Required Parameters****cloudType**

- **type:** string
- **description:** Cloud type
- **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

**paymentOption**

- **type:** string
- **description:** Preferred payment option

- **enum:** ['ALL\_UPFRONT', 'PARTIAL\_UPFRONT', 'NO\_UPFRONT']

#### **reservedInstanceType**

- **type:** string
- **description:** Preferred offering class (Standard or Convertible)
- **enum:** ['STANDARD', 'CONVERTIBLE']

#### **termYears**

- **type:** integer
- **format:** int32
- **description:** Number of years in preferred term

## **QueryInputApiDTO**

### **Description**

Model to describe an input query

### **Required Parameters**

#### **query**

- **type:** string
- **description:** Query string. The format of this depends on the QueryType chosen. For example, for a REGEX QueryType, this should be a regular expression.

#### **type**

- **type:** string
- **description:** Query type
- **enum:** ['EXACT', 'REGEX', 'CONTAINS']

### **Optional Parameters**

#### **caseSensitive**

- **type:** boolean
- **description:** If true, the query string will match case-sensitivity. This field is only used when QueryType is not REGEX.

## **RIBuyActionDetailsApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [ActionDetailsApiDTO](#)

### **Optional Parameters**

#### **estimatedOnDemandCost**

- **type:** number
- **format:** float



### **estimatedRICoverage**

- **type:** number
- **format:** float

### **historicalDemandDataset**

- **type:** array
- – **StatSnapshotApiDTO**

### **subType**

- **type:** string

## **RIPurchaseProfilesSettingApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [SettingApiDTO](#)

### **Required Parameters**

#### **value**

- **description:** Value of the RI Purchase Profiles of type ReservedInstancePurchaseSettingsDTO.
- **ReservedInstancePurchaseSettingsDTO**

### **Optional Parameters**

#### **defaultValue**

- **description:** Default value before any change
- **readOnly:** True
- **ReservedInstancePurchaseSettingsDTO**

## **RangeApiDTO**

### **Description**

Model to describe the Range of a Value in the Settings

### **Optional Parameters**

#### **step**

- **type:** number
- **format:** double
- **description:** The Step scale factor of the slider

#### **labels**

- **type:** array
- **description:** Labels to show in the values of the slider
- – **type:** string

### **customStepValues**

- **type:** array
- **description:** Custom step values when the slider step is not constant - [DEPRECATED - Use: stepValues]
- – **type:** integer
  - **format:** int32

### **stepValues**

- **type:** array
- **description:** Custom step values when the slider step is not constant
- – **type:** number
  - **format:** float

## **RangeInputApiDTO**

### **Description**

Model to describe an input range. Either minValue or maxValue needs to be defined.

### **Optional Parameters**

#### **minValue**

- **type:** number
- **format:** float
- **description:** Minimum value of the input range

#### **maxValue**

- **type:** number
- **format:** float
- **description:** Maximum value of the input range

## **RateCardApiDTO**

### **Description**

Model to describe a ratecard.

### **Required Parameters**

#### **info**

- **description:** List of business unit UUID's related to this rate card.
- [RateCardInfoApiDTO](#)

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string

- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**type**

- **type:** string
- **description:** Type of rate card", example = "AWS, Azure"
- **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

**related**

- **description:** List of business units related to this rate card.
- **RateCardRelatedApiDTO**

**validation**

- **description:** Rate card validation.
- **RateCardValidationApiDTO**

**valid**

- **type:** boolean

**default**

- **type:** boolean

**fileName**

- **type:** string

## RateCardInfoApiDTO

### Description

Model to describe the attributes of a rate card.

### Required Parameters

**businessUnitsIds**

- **type:** array
- **description:** List of business units associated to this rate card.
- – **type:** string

## RateCardRelatedApiDTO

### Description

Model to describe the related business units of this ratecard.

### Optional Parameters

**businessUnits**

- **type:** array
- **description:** List of business units related to the ratecard
- – **BusinessUnitApiDTO**

## RateCardValidationApiDTO

### Description

Model to validate the RateCard

### Optional Parameters

#### name

- **type:** string
- **description:** Name of the rate card file

#### status

- **type:** string
- **example:** VALID, INVALID
- **description:** Validation Status
- **enum:** ['VALID', 'INVALID']

#### message

- **type:** string
- **example:** Validation was successful, Validation failed
- **description:** Reason for validation failure or success.

## Recommendation

### Description

A DTO for a single recommendation to an error.

### Required Parameters

#### description

- **type:** string
- **description:** The recommended user action to investigate/mitigate the problem.

#### errorType

- **type:** string
- **description:** The problem this recommendation is produced for.

## ReconfigureActionDetailsApiDTO

### Description

### Inherits From

This DTO inherits from [ActionDetailsApiDTO](#)

### Optional Parameters

#### vcpuBefore

- **type:** integer

- **format:** int32
- **description:** vCPUs before the change

#### **vcpuAfter**

- **type:** integer
- **format:** int32
- **description:** vCPUs after the change

#### **socketsBefore**

- **type:** integer
- **format:** int32
- **description:** sockets before the change

#### **socketsAfter**

- **type:** integer
- **format:** int32
- **description:** sockets after the change

#### **coresPerSocketBefore**

- **type:** integer
- **format:** int32
- **description:** cores per socket before the change

#### **coresPerSocketAfter**

- **type:** integer
- **format:** int32
- **description:** cores per socket after the change

## **RecurrenceApiDTO**

### **Description**

Model to describe the Schedule for a Settings Policy

### **Required Parameters**

#### **type**

- **type:** string
- **description:** Type of the recurrence, Enum: DAILY, WEEKLY, MONTHLY
- **enum:** ['DAILY', 'WEEKLY', 'MONTHLY']

### **Optional Parameters**

#### **daysOfWeek**

- **type:** array
- **description:** The day of week that the recurrence will occur. Required if the type of recurrence is WEEKLY, or if the type is MONTHLY and a week for the monthly recurrence has been set.
- – **type:** string
  - **enum:** ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']

### daysOfMonth

- **type:** array
- **description:** When recurrence type is MONTHLY, the date to apply the monthly recurrence
- – **type:** integer
  - **format:** int32

### weekOfTheMonth

- **type:** array
- **description:** When recurrence type is MONTHLY, which week of the month to apply the recurrence. A value of -1 indicates last week, 0 indicates this week, and so on.
- – **type:** integer
  - **format:** int32

### interval

- **type:** integer
- **format:** int32
- **description:** Frequency of the recurrence based on the recurrence type. For example, an interval of '2' with a recurrence type of MONTHLY will occur every two months.

## RegionAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### latitude

- **type:** number
- **format:** double
- **description:** The latitude of the region used along with longitude to determine physical location.

#### longitude

- **type:** number
- **format:** double
- **description:** The longitude of the region used along with latitude to determine physical location.

#### numWorkloads

- **type:** integer
- **format:** int32
- **description:** Number of workloads (VMs, databases, etc.) in this region.

## RelatedActionApiDTO

### Description

Model to describe the relationship between two actions

## Optional Parameters

### action

- **description:** The basic action DTO for the related action
- **BasicActionApiDTO**

### actionRelationType

- **type:** string
- **description:** Action relation type
- **enum:** ['BLOCKED\_BY', 'BLOCKING', 'CAUSED\_BY', 'CAUSING']

## RelatedEntityFieldApiDTO

### Description

### Inherits From

This DTO inherits from [FieldApiDTO](#)

### Required Parameters

#### property

- **type:** string
- **description:** The related entity property of interest.
- **readOnly:** True
- **enum:** ['NAMES', 'COUNT']

#### relatedEntityType

- **type:** string
- **description:** The type of related entities of interest.
- **readOnly:** True
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

## RelatedGroupFieldApiDTO

### Description

### Inherits From

This DTO inherits from [FieldApiDTO](#)

## Required Parameters

### property

- **type:** string
- **description:** The related group property of interest.
- **readOnly:** True
- **enum:** ['NAMES', 'COUNT']

### relatedGroupType

- **type:** string
- **description:** The type of related groups of interest.
- **readOnly:** True
- **enum:** ['Group', 'Resource', 'Cluster', 'VMCluster', 'StorageCluster', 'BusinessAccountFolder', 'BillingFamily', 'NodePool']

## RelievePressureObjectApiDTO

### Description

Model to describe a relieve pressure change in the scenario: minimum amount of Actions Moves that will relieve the pressure on the over-utilized clusters

### Optional Parameters

#### projectionDay

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### sources

- **type:** array
- **description:** Sources of the change, List of Clusters that are overloaded
- – [BaseApiDTO](#)

#### destinations

- **type:** array
- **description:** Destinations of the change, List of Clusters where VMs can move to
- – [BaseApiDTO](#)

## RemoveConstraintApiDTO

### Description

Model to describe a constraint change in the scenario

### Optional Parameters

#### constraintType

- **type:** string
- **description:** Type of the constraint to remove. To ignore all constraints - use AllCommodities. This will ignore all constraints not just Network/Cluster/StorageCluster/DC. GlobalIgnoreConstraint - deprecated - use AllCommodities instead.



- **enum:** ['DataCenterCommodity', 'ClusterCommodity', 'NetworkCommodity', 'DatastoreCommodity', 'StorageClusterCommodity', 'GlobalIgnoreConstraint', 'AllCommodities']

#### **projectionDay**

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### **target**

- **description:** Target of the change, when not provided change applied to plan scope
- **BaseApiDTO**

#### **targetEntityType**

- **type:** string
- **description:** EntityType to target with constraint configuration
- **enum:** ['VirtualMachine', 'Container', 'ContainerPod']

## **RemoveObjectApiDTO**

### **Description**

Model to describe a remove change in the scenario

### **Optional Parameters**

#### **projectionDay**

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### **targetEntityType**

- **type:** string
- **description:** Type of entity to remove from group target

#### **target**

- **description:** Target of the change, entity or group
- **BaseApiDTO**

## **ReplaceObjectApiDTO**

### **Description**

Model to describe a migration change in the scenario

### **Optional Parameters**

#### **projectionDay**

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

### target

- **description:** Target of the change, entity or group
- **BaseApiDTO**

### template

- **description:** Template used for the replace
- **BaseApiDTO**

### targetEntityType

- **type:** string
- **description:** Type of entity to remove from group target

## RequestHeader

### Description

### Required Parameters

#### name

- **type:** string
- **description:** Request header name

#### value

- **type:** string
- **description:** Request header value

## ReservationConstraintApiDTO

### Description

The constraints user specified when running reservation which limit where templates could be placed.

### Required Parameters

#### constraintType

- **type:** string
- **description:** Type of constraint
- **enum:** ['CLUSTER', 'DATA\_CENTER', 'VIRTUAL\_DATA\_CENTER', 'POLICY', 'NETWORK']

#### uuid

- **type:** string
- **description:** UUID of the constraint

### Optional Parameters

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

## ReservationFailureInfoDTO

### Description

Failure information when reservation fails

### Required Parameters

#### closestSeller

- **description:** provider that came closest to meeting the requirements
- **BaseApiDTO**

#### closestSellerCluster

- **description:** Closest provider's cluster information
- **BaseApiDTO**

#### maxQuantityAvailable

- **type:** number
- **format:** double
- **description:** Quantity of resource available on closestSeller

#### quantityRequested

- **type:** number
- **format:** double
- **description:** Quantity of resource requested

#### resource

- **type:** string
- **description:** Most constrained resource

#### timespan

- **type:** string
- **description:** Failure is in historical or current
- **enum:** ['HISTORICAL', 'CURRENT', 'PROJECTED', 'PLAN\_SOURCE', 'PLAN\_PROJECTED']

### Optional Parameters

#### units

- **type:** string
- **description:** Units for the resource

## ReservationInvalidInfoApiDTO

### Description

The reason the reservation is invalid

## Optional Parameters

### invalidReason

- **type:** string
- **description:** The reason why the reservation is invalid
- **enum:** ['MARKET\_CONNECTIVITY\_ERROR', 'MARKET\_NOT\_READY', 'INVALID\_CONSTRAINTS']

## ReservedInstanceApiDTO

### Description

Model to describe a Reserved Instance

### Required Parameters

#### cloudType

- **type:** string
- **description:** Cloud Type
- **enum:** ['AWS', 'AZURE', 'GCP', 'HYBRID', 'UNKNOWN']

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### template

- **description:** Related template
- **BaseApiDTO**

#### platform

- **type:** string
- **description:** Platform name
- **enum:** ['UNKNOWN', 'LINUX', 'RHEL', 'SUSE', 'WINDOWS', 'WINDOWS\_WITH\_SQL\_STANDARD', 'WINDOWS\_WITH\_SQL\_WEB', 'WINDOWS\_WITH\_SQL\_SERVER\_ENTERPRISE', 'LINUX\_WITH\_SQL\_STANDARD', 'LINUX\_WITH\_SQL\_WEB', 'LINUX\_WITH\_SQL\_SERVER\_ENTERPRISE']

#### type

- **type:** string
- **description:** Reserved instance type
- **enum:** ['STANDARD', 'CONVERTIBLE']

#### location

- **description:** Location, either a zone or a data center if zone is not available
- **BaseApiDTO**

**accountId**

- **type:** string
- **description:** Account ID

**accountDisplayName**

- **type:** string
- **description:** Account display name

**targetId**

- **type:** string
- **description:** The associated target ID of the account

**masterAccountId**

- **type:** string
- **description:** Master account ID

**term**

- **description:** Term, unit is Year
- **StatApiDTO**

**payment**

- **type:** string
- **description:** Payment option
- **enum:** ['ALL\_UPFRONT', 'PARTIAL\_UPFRONT', 'NO\_UPFRONT']

**onDemandPrice**

- **description:** On-demand price, saving is RI price minus on-demand price
- **StatApiDTO**

**costPrice**

- **description:** Total monthly cost
- **StatApiDTO**

**coupons**

- **description:** Total coupons and used coupons, utilization = numOfCouponsUsed / numOfCoupons
- **StatApiDTO**

**totalCoupons**

- **type:** number
- **format:** float
- **description:** Total coupons provided by the cbtp the vm matched with

**expDate**

- **type:** string
- **description:** Expiration date

**instanceCount**

- **type:** integer
- **format:** int32
- **description:** Number of reserved instance with the same type, template, platform etc..

**tenancy**

- **type:** string
- **description:** Tenancy
- **enum:** ['DEFAULT', 'DEDICATED', 'HOST']

**upFrontCost**

- **type:** number
- **format:** double
- **description:** Up front cost

**actualHourlyCost**

- **type:** number
- **format:** double
- **description:** Actual hourly cost

**effectiveHourlyCost**

- **type:** number
- **format:** double
- **description:** Effective hourly cost

**isExpired**

- **type:** boolean
- **description:** Whether the RI be expired by a date provided

**expDateEpochTime**

- **type:** integer
- **format:** int64
- **description:** Expiration Date in epoch time

**scopeType**

- **type:** string
- **description:** Whether the RI scope is Single or Shared (Azure)
- **enum:** ['SINGLE', 'SHARED', 'UNKNOWN']

**appliedScopes**

- **type:** array
- **description:** List of applied scope UUIDs- only provided when scope is Single (Azure)
- – **type:** string

**orderID**

- **type:** string
- **description:** The Order ID of the RI (Azure)

**trueID**

- **type:** string
- **description:** The ID of the RI

**coveredEntityCount**

- **type:** integer
- **format:** int32
- **description:** Count of workload entities covered by the reserved instance

**undiscoveredAccountsCoveredCount**

- **type:** integer
- **format:** int32
- **description:** Count of undiscovered accounts covered by the reserved instance

**sizeFlexible**

- **type:** boolean

**toBuy**

- **type:** boolean

## ReservedInstancePurchaseSettingsDTO

**Description****Optional Parameters****awsProfile**

- PurchaseProfileDTO

**azureProfile**

- PurchaseProfileDTO

## ResizeActionDetailsApiDTO

**Description**

Parent class for resize action details. Implementing classes include: CloudResizeActionDetailsApiDTO.

**Required Parameters****actionDetailType**

- **type:** string

**actionType**

- **type:** string

**Optional Parameters****subType**

- **type:** string

## ResourceApiDTO

**Description**

Model to describe the Resources of a Template

**Optional Parameters****stats**

- **type:** array

- **description:** Statistics, e.g. Number of CPU, Memory size, Disk size, ...

- – **StatApiDTO**

#### **provider**

- **description:** Entity provider

- **BaseApiDTO**

#### **type**

- **type:** string

- **description:** Type of the resource

#### **template**

- **type:** string

- **description:** Template used to fulfill the resources

#### **relatedResources**

- **type:** array

- **description:** ID of resources related to this resource

- – **type:** string

#### **linkedResources**

- **type:** array

- **description:** resources related to this resource. For now, this will only contain the cluster

- – **BaseApiDTO**

## **ResourceGroupApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [GroupApiDTO](#)

### **Optional Parameters**

#### **parentUuid**

- **type:** string

- **description:** Parent Business Account uuid (e.g subscription in Azure) where the Resource Group belongs to.

#### **parentDisplayName**

- **type:** string

- **description:** Parent Business Account display name (e.g subscription in Azure) where the Resource Group belongs to.

## **RoleApiDTO**

### **Description**

Model to describe a User Role, it describe the permissions of an user over the Ops. Manager capabilities



## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### name

- **type:** string
- **description:** Name of the role

## SAMLIdpApiDTO

### Description

Model to describe an SAML IDP. It contains IDP URL and is SAML ONLY enabled or not.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### idpURL

- **type:** string
- **description:** The SAML IDP URL

### isSAMLOnly

- **type:** boolean

### isSingleLogoutEnabled

- **type:** boolean

## STEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

## Optional Parameters

### displayName

- **type:** string
- **description:** Display Name

### name

- **type:** string
- **description:** Name

### productName

- **type:** string
- **description:** Product Name

### productType

- **type:** string
- **description:** Product Type

### deviceId

- **type:** string
- **description:** Device ID

### deviceName

- **type:** string
- **description:** Device Name

### externalNames

- **type:** array
- **description:** External Names
- – **type:** string

### altName

- **type:** string
- **description:** Alternative Name

### ssdDiskCount

- **type:** integer
- **format:** int64
- **description:** SSD Disk Count

### rpm7200DiskCount

- **type:** integer
- **format:** int64
- **description:** RPM 7200 Disk Count

### rpm10KDiskCount

- **type:** integer
- **format:** int64
- **description:** RPM 10K Disk Count

### rpm15KDiskCount

- **type:** integer

- **format:** int64
- **description:** RPM 15K Disk Count

#### **getvSeriesDiskCount**

- **type:** integer
- **format:** int64
- **description:** VSeries Disk Count

#### **ssdDiskIOPSCapacity**

- **type:** number
- **format:** double
- **description:** SSD Disk IOPS Capacity

#### **rpm7200DiskIOPSCapacity**

- **type:** number
- **format:** double
- **description:** RPM 7200 Disk IOPS Capacity

#### **rpm10KDiskIOPSCapacity**

- **type:** number
- **format:** double
- **description:** RPM 10K Disk IOPS Capacity

#### **rpm15KDiskIOPSCapacity**

- **type:** number
- **format:** double
- **description:** RPM 15K Disk IOPS Capacity

#### **getvSeriesDiskIOPSCapacity**

- **type:** number
- **format:** double
- **description:** VSeries Disk IOPS Capacity

#### **technologyType**

- **type:** string
- **description:** HCI Technology Type

#### **rawCapacity**

- **type:** integer
- **format:** int64
- **description:** HCI Raw Capacity

#### **rawFreespace**

- **type:** integer
- **format:** int64
- **description:** HCI Raw Freespace

#### **rawUncommitted**

- **type:** integer
- **format:** int64
- **description:** HCI Raw Uncommitted

**redundancyMethod**

- **type:** string
- **description:** HCI Redundancy Method

**failuresToTolerate**

- **type:** integer
- **format:** int32
- **description:** HCI Failures To Tolerate

**spaceReservationPct**

- **type:** integer
- **format:** int32
- **description:** HCI Space Reservation Percentage

## ScenarioApiDTO

### Description

Model to describe a What-If Scenario, composed by a list of changes

### Optional Parameters

**uuid**

- **type:** string

**displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**owners**

- **type:** array
- **description:** Users that owns the scenario
- – [UserApiDTO](#)

**type**

- **type:** string
- **description:** Type of the scenario

**scope**

- **type:** array
- **description:** List of entities/group that define the Scope
- – [BaseApiDTO](#)

**projectionDays**

- **type:** array
- **description:** List of days that define when the scenario should run
- – **type:** integer
- **format:** int32

### topologyChanges

- **description:** Changes to apply to the topology of the plan market
- **TopologyChangesApiDTO**

### loadChanges

- **description:** Workload utilization changes in the scenario
- **LoadChangesApiDTO**

### configChanges

- **description:** Configuration changes in the scenario
- **ConfigChangesApiDTO**

### timebasedTopologyChanges

- **TimeBasedTopologyChangesApiDTO**

## ScheduleApiDTO

### Description

Model to describe the Schedule for a Settings Policy

### Required Parameters

#### endTime

- **type:** string
- **format:** date-time
- **example:** 2019-04-04T19:15
- **description:** End time in hours and minutes for the scheduled period in ISO8601. For example, 2019-04-04T19:15

#### startDate

- **type:** string
- **format:** date-time
- **example:** 2019-04-04T19:15
- **description:** Start date of the schedule period, expressed using ISO8601 Format (YYYY-MM-DDTHH:MM:SS). This date is local to the timezone of the instance.

#### startTime

- **type:** string
- **format:** date-time
- **example:** 2019-04-04T19:15
- **description:** Start time in hours and minutes for the scheduled period in ISO8601. For example, 2019-04-04T17:15

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string

- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**endDate**

- **type:** string
- **format:** date
- **example:** 2019-04-04
- **description:** End date of the schedule period, expressed using ISO8601 Format (YYYY-MM-DDTHH:MM:SS). This date is local to the timezone of the instance.

**recurrence**

- **description:** Recurrence setup
- **RecurrenceApiDTO**

**timeZone**

- **type:** string
- **description:** The time zone name based on tz database

**nextOccurrence**

- **type:** string
- **description:** The date and time of the next occurrence of this schedule represented in ISO8601 local time (YYYY-MM-DDTHH:MM:SS). For example, 2019-04-04T19:15

**nextOccurrenceTimestamp**

- **type:** integer
- **format:** int64
- **description:** The date and time of the next occurrence of this schedule represented in Timestamp in milliseconds. For example, 1562100619000

**remaingTimeActiveInMs**

- **type:** integer
- **format:** int64
- **description:** The amount of time that the schedule is going to be active at the time of call in milliseconds. This field will not be populated if the schedule is not currently active.

## ScheduleEntityResponseApiDTO

### Description

represents a response of the attach/detach time span schedule to/from the parkable entity

### Optional Parameters

**error**

- **type:** string
- **example:** entityUuid not found
- **description:** error indicating the reason for failure

**entityUUID**

- **type:** string

## ScheduleItemApiDTO

### Description

Base model to describe a ScheduleItem.

### Required Parameters

#### type

- **type:** string
- **example:** ParkingItemApiDTO
- **description:** ScheduleItem type discriminator. Describes which subclass of ScheduleItem is returned.

## ScheduleTimeSpansApiDTO

### Description

Parking time span schedule

### Required Parameters

#### timeZone

- **type:** string
- **example:** America/New\_York
- **description:** Timezone for the time spans in this schedule. Should be chosen from the IANA Timezone Database here: [<https://www.iana.org/time-zones>]. See the list in wikipedia here: [[https://en.wikipedia.org/wiki/List\\_of\\_tz\\_database\\_time\\_zones](https://en.wikipedia.org/wiki/List_of_tz_database_time_zones)]

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### description

- **type:** string
- **example:** Up 7-5 on weekdays
- **description:** Description of the schedule

#### timeSpans

- **description:** Weekly schedule time spans
- **TimeSpanGroupApiDTO**

## ScopeUuidsApiInputDTO

### Description

Model to describe the request by UUIDs

### Required Parameters

#### uuids

- **type:** array
- **description:** List of uuids
- – **type:** string
- **maxItems:** 2147483647
- **minItems:** 1

### Optional Parameters

#### marketId

- **type:** string
- **description:** Market ID associated with the requested items

## SearchAllQueryApiDTO

### Description

A query used to search across the entire topology (entities and groups)

### Required Parameters

#### select

- **description:** Defines what subset of types (entities and groups) should be returned.
- **readOnly:** True
- **SelectAllApiDTO**

#### where

- **description:** Conditions that each retrieved object must satisfy (currently only name is supported).
- **readOnly:** True
- **WhereApiDTO**

### Optional Parameters

#### pagination

- **description:** Pagination information.
- **readOnly:** True
- **PaginationApiDTO**

## SearchCountRecordApiDTO

### Description

A count of entities.



## Required Parameters

### count

- **type:** integer
- **format:** int32
- **description:** Count of entities.
- **readOnly:** True

### groupBy

- **type:** array
- **description:** List of fields describing the groupBy that this count represents
- **readOnly:** True
- – [FieldValueApiDTO](#)

## SearchQueryRecordApiDTO

### Description

A record with information about an entity or a group.

## Required Parameters

### oid

- **type:** integer
- **format:** int64
- **description:** Unique identifier for the entity or group.
- **readOnly:** True

### values

- **type:** array
- **description:** Information about the entity or group.
- **readOnly:** True
- – [FieldValueApiDTO](#)

## SelectAllApiDTO

### Description

Specification of what subset of types (entities and groups) should be returned.

## Optional Parameters

### entityTypes

- **type:** array
- **description:** Entity types to include in the response.
- **readOnly:** True
- – **type:** string
  - **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray',

'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

### groupTypes

- **type:** array
- **description:** Group types to include in the response.
- **readOnly:** True
- – **type:** string
  - **enum:** ['Group', 'Resource', 'Cluster', 'VMCluster', 'StorageCluster', 'BusinessAccountFolder', 'BillingFamily', 'NodePool']

## SelectEntityApiDTO

### Description

Specification of what fields should be fetched in an entity search response.

### Required Parameters

#### entityType

- **type:** string
- **description:** Entity Type
- **readOnly:** True
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

### Optional Parameters

#### fields

- **type:** array
- **description:** Fields to fetch.
- **readOnly:** True
- – **FieldApiDTO**

## SelectGroupApiDTO

### Description

Specification of what fields should be fetched in a group search response.

## Required Parameters

### groupType

- **type:** string
- **description:** Group Type.
- **readOnly:** True
- **enum:** ['Group', 'Resource', 'Cluster', 'VMCluster', 'StorageCluster', 'BusinessAccountFolder', 'BillingFamily', 'NodePool']

## Optional Parameters

### fields

- **type:** array
- **description:** Fields to fetch.
- **readOnly:** True
- – **FieldApiDTO**

## ServiceEntityApiDTO

### Description

Model to describe an Entity, e.g. Virtual Machine, Physical Machine, Storage

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

### discoveredBy

- **description:** Target that discovered the entity
- **TargetApiDTO**

### vendorIds

- **type:** object
- **description:** The mapping of target identifier to vendor-provided identity of this entity on the remote target
- **additionalProperties:** {'type': 'string'}

### priceIndex

- **type:** number
- **format:** float
- **description:** Price index of the Entity

### state

- **type:** string
- **description:** State
- **enum:** ['IDLE', 'RESOURCE\_ALLOCATION', 'LAUNCH', 'ACTIVE', 'RESOURCE\_RELEASE', 'SUSPEND', 'SUSPEND\_PENDING', 'TERMINATE\_PENDING', 'NOT\_MONITORED', 'MAINTENANCE', 'FAILOVER', 'UNKNOWN', 'EVACUATED', 'QUEUED']

### severity

- **type:** string
- **description:** Highest severity of the Actions related to this entity
- **enum:** ['UNKNOWN', 'NORMAL', 'MINOR', 'MAJOR', 'CRITICAL']

### unplacementDetails

- **description:** Additional information provided for unplaced entities in a plan
- **UnplacementDetailsApiDTO**

### placedOn

- **type:** string
- **description:** The comma-separated classNames of entities successfully placed on, used for Unplaced entities in a Plan Market - [DEPRECATED - Use: unplacementDetails.placedOn]

### notPlacedOn

- **type:** string
- **description:** The comma-separated classNames of entities could not be placed on, used for Unplaced entities in a Plan Market - [DEPRECATED - Use: unplacementDetails.reasons]

### unplacedExplanation

- **type:** string
- **description:** The reason(s) the entity could not be placed - [DEPRECATED - Use: unplacementDetails.reasons]

### costPrice

- **type:** number
- **format:** float
- **description:** Cost price per Hour

### severityBreakdown

- **type:** object
- **description:** A Map of severity along with their count. E.g. {CRITICAL: 1, NORMAL: 10}
- **additionalProperties:** {'type': 'integer', 'format': 'int64'}

### remoteId

- **type:** string
- **description:** The identifier of this instance on the remote target from which it was discoveredc - [DEPRECATED - Use: vendorIds]

### **realtimeMarketReference**

- **description:** In case of a Plan Market Entity, this contains the original entity from the Real Time Market
- **BaseApiDTO**

### **providers**

- **type:** array
- **description:** List of Providers
- – **BaseApiDTO**

### **consumers**

- **type:** array
- **description:** List of Consumers
- – **BaseApiDTO**

### **template**

- **description:** Related Template
- **TemplateApiDTO**

### **numRelatedVMs**

- **type:** integer
- **format:** int32
- **description:** Number of related VMs

### **connectedEntities**

- **type:** array
- **description:** Limited list of connected entities relevant to the service entity. This will not contain all connected entities.
- – **BaseApiDTO**

### **aspects**

- **type:** object
- **description:** Additional info about the Entity categorized as Aspects
- **additionalProperties:** {'\$ref': '#/definitions/EntityAspect'}

### **tags**

- **type:** object
- **description:** Tags are the metadata defined in name/value pairs. Each name can have multiple values.
- **additionalProperties:** {'type': 'array', 'items': {'type': 'string'}}

### **onDemandRateBefore**

- **type:** number
- **format:** float
- **description:** On-demand rate before actions

### **onDemandRateAfter**

- **type:** number
- **format:** float
- **description:** On-demand rate after actions

### staleness

- **type:** string
- **description:** [BETA]: Whether the entity is outdated, i.e. discovered too long ago to be considered for analytical purposes
- **enum:** ['CURRENT', 'STALE']

## ServletOutputStream

### Description

### Optional Parameters

#### ready

- **type:** boolean

## SettingActivePolicyApiDTO

### Description

Model to describe a Settings Policy that affect the value of a Setting

### Required Parameters

#### numEntities

- **type:** integer
- **format:** int32
- **description:** Number of entities affected by the Settings Policy value

#### settingsPolicy

- **description:** Settings Policy that affects the final value of a Setting
- **BaseApiDTO**

#### value

- **type:** string
- **description:** Setting value declared in the Settings Policy

## SettingApiDTOSerializable

### Description

Generic model to describe a Setting. Sub types include StringSettingApiDTO and RIPurchaseProfilesSettingApiDTO.

### Required Parameters

#### value

- **description:** Value of the setting. Type indicated by the `valueObjectType` field. The default type is `String`
- **\$ref:** [#/definitions/object](#)

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### valueDisplayName

- **type:** string
- **description:** Display Name of the Value, in case the value is a Template or another entity
- **readOnly:** True

### defaultValue

- **description:** Default value before any change
- **readOnly:** True
- **\$ref:** [#/definitions/object](#)

### categories

- **type:** array
- **description:** Category and sub categories of the setting
- **readOnly:** True
- – **type:** string

### valueType

- **type:** string
- **description:** Define the type of the value
- **enum:** ['STRING', 'NUMERIC', 'INTEGER', 'BOOLEAN']

### valueObjectType

- **type:** string
- **description:** The type of complex object values. Defaults to 'String'.
- **enum:** ['String', 'ReservedInstancePurchaseSettingsDTO']

### options

- **type:** array
- **description:** List of allowable value for this setting
- **readOnly:** True
- – **SettingOptionApiDTO**

### min

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the min limit
- **readOnly:** True

#### max

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the max limit
- **readOnly:** True

#### entityType

- **type:** string
- **description:** Entity type for this Setting

#### range

- **description:** Range options for the value of the setting
- **readOnly:** True
- **RangeApiDTO**

#### scope

- **type:** string
- **description:** Define the visibility of the setting. E.G. a Local Setting is only visible when associated with an Entity/Group
- **readOnly:** True
- **enum:** ['GLOBAL', 'LOCAL']

#### activeSettingsPolicies

- **type:** array
- **description:** List of Settings Policies that are affecting the final value of this setting
- – **SettingActivePolicyApiDTO**

#### sourceGroupName

- **type:** string
- **description:** Source group name for setting value

#### sourceGroupUuid

- **type:** string
- **description:** Source group uuid for setting value

## SettingApiDTOString

### Description

Generic model to describe a Setting. Sub types include StringSettingApiDTO and RIPurchaseProfilesSettingApiDTO.

### Required Parameters

#### value

- **description:** Value of the setting. Type indicated by the `valueObjectType` field. The default type is `String`
- **\$ref:** [#/definitions/object](#)



## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### valueDisplayName

- **type:** string
- **description:** Display Name of the Value, in case the value is a Template or another entity
- **readOnly:** True

### defaultValue

- **description:** Default value before any change
- **readOnly:** True
- **\$ref:** [#/definitions/object](#)

### categories

- **type:** array
- **description:** Category and sub categories of the setting
- **readOnly:** True
- – **type:** string

### valueType

- **type:** string
- **description:** Define the type of the value
- **enum:** ['STRING', 'NUMERIC', 'INTEGER', 'BOOLEAN']

### valueObjectType

- **type:** string
- **description:** The type of complex object values. Defaults to 'String'.
- **enum:** ['String', 'ReservedInstancePurchaseSettingsDTO']

### options

- **type:** array
- **description:** List of allowable value for this setting
- **readOnly:** True
- – **SettingOptionApiDTO**

### min

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the min limit
- **readOnly:** True

### max

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the max limit
- **readOnly:** True

### entityType

- **type:** string
- **description:** Entity type for this Setting

### range

- **description:** Range options for the value of the setting
- **readOnly:** True
- **RangeApiDTO**

### scope

- **type:** string
- **description:** Define the visibility of the setting. E.G. a Local Setting is only visible when associated with an Entity/Group
- **readOnly:** True
- **enum:** ['GLOBAL', 'LOCAL']

### activeSettingsPolicies

- **type:** array
- **description:** List of Settings Policies that are affecting the final value of this setting
- – **SettingActivePolicyApiDTO**

### sourceGroupName

- **type:** string
- **description:** Source group name for setting value

### sourceGroupUuid

- **type:** string
- **description:** Source group uuid for setting value

## SettingApiDTO

### Description

Generic model to describe a Setting. Sub types include StringSettingApiDTO and RIPurchaseProfilesSettingApiDTO.

### Required Parameters

#### value

- **description:** Value of the setting. Type indicated by the `valueObjectType` field. The default type is `String`
- **\$ref:** [#/definitions/object](#)

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### valueDisplayName

- **type:** string
- **description:** Display Name of the Value, in case the value is a Template or another entity
- **readOnly:** True

### defaultValue

- **description:** Default value before any change
- **readOnly:** True
- **\$ref:** [#/definitions/object](#)

### categories

- **type:** array
- **description:** Category and sub categories of the setting
- **readOnly:** True
- – **type:** string

### valueType

- **type:** string
- **description:** Define the type of the value
- **enum:** ['STRING', 'NUMERIC', 'INTEGER', 'BOOLEAN']

### valueObjectType

- **type:** string
- **description:** The type of complex object values. Defaults to 'String'.
- **enum:** ['String', 'ReservedInstancePurchaseSettingsDTO']

### options

- **type:** array
- **description:** List of allowable value for this setting
- **readOnly:** True
- – **SettingOptionApiDTO**

### min

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the min limit
- **readOnly:** True

### max

- **type:** number
- **format:** double
- **description:** If type is NUMERIC, this define the max limit
- **readOnly:** True

### entityType

- **type:** string
- **description:** Entity type for this Setting

### range

- **description:** Range options for the value of the setting
- **readOnly:** True
- **RangeApiDTO**

### scope

- **type:** string
- **description:** Define the visibility of the setting. E.G. a Local Setting is only visible when associated with an Entity/Group
- **readOnly:** True
- **enum:** ['GLOBAL', 'LOCAL']

### activeSettingsPolicies

- **type:** array
- **description:** List of Settings Policies that are affecting the final value of this setting
- – **SettingActivePolicyApiDTO**

### sourceGroupName

- **type:** string
- **description:** Source group name for setting value

### sourceGroupUuid

- **type:** string
- **description:** Source group uuid for setting value

## SettingOptionApiDTO

### Description

### Optional Parameters

#### label

- **type:** string

#### value

- **type:** string

## SettingsManagerApiDTO

### Description

Model to describe a Category Manager for the settings

### Required Parameters

#### category

- **type:** string
- **description:** Category of the Manager

#### settings

- **type:** array
- **description:** List of Settings contained in the Manager
- – SettingApiDTOSerializable

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

## SettingsPolicyApiDTO

### Description

Model to describe a Setting Policy: a group of Settings that are applied based on Scope and/or Schedule

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### entityType

- **type:** string
- **description:** Type of the Settings Policy. E.G. VirtualMachine, PhysicalMachine, ...

### scopes

- **type:** array
- **description:** List of scopes where the settings are applied, on create/edit, only Uuid is required. If null it's considered Global scope
- – **GroupApiDTO**

### settingsManagers

- **type:** array
- **description:** List of groups of Settings to activate in this Policy
- – **SettingsManagerApiDTO**

### schedule

- **description:** Schedule to define when the settings are applied
- **ScheduleApiDTO**

### disabled

- **type:** boolean
- **description:** Control if the policy is Enabled or Disabled

### note

- **type:** string
- **description:** Note for this Policy

### readOnly

- **type:** boolean
- **description:** If true, prevents users from deleting or modifying the SettingPolicy.

### default

- **type:** boolean

## SkippedActionApiDTO

### Description

Model to describe an action that was skipped in execution.

### Required Parameters

#### actionId

- **type:** string
- **description:** Action ID

### Optional Parameters

#### reason

- **type:** string
- **description:** Reason why action was skipped

# StatApiDTO

## Description

Model to describe a Statistic

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### name

- **type:** string
- **description:** Name

### capacity

- **description:** Capacity values, used for Commodities stats
- [StatValueApiDTO](#)

### reserved

- **description:** Reserved values, used for Commodities stats
- [StatValueApiDTO](#)

### relatedEntityType

- **type:** string
- **description:** E.G.virtual machine

### filters

- **type:** array
- **description:** describe the grouping options used to generate the output
- – [StatFilterApiDTO](#)

### relatedEntity

- **description:** Provider or Consumer of the Statistic, 'relatedEntity' and 'numRelatedEntities' are mutually exclusive
- [BaseApiDTO](#)

### numRelatedEntities

- **type:** integer
- **format:** int32
- **description:** Number of providers or consumers for this statistic, 'relatedEntity' and 'numRelatedEntities' are mutually exclusive

### units

- **type:** string

- **description:** Units, used for Commodities stats. E.G. \$/h

#### values

- **description:** Contains Average, Min, Max, Total
- **StatValueApiDTO**

#### value

- **type:** number
- **format:** float
- **description:** Simple value, equal to values.avg.

#### commoditySource

- **description:** Source for a commodity's capacity and usage, used for Commodities stats
- **StatCommoditySourceApiDTO**

#### histUtilizations

- **type:** array
- **description:** Commodity historical utilization values
- – **StatHistUtilizationApiDTO**

## StatApiInputDTO

### Description

Model to describe the request of a Stat. Please note that the costPrice stat is deprecated and will EOL in 8.10.1. Use the dedicated 'cost' endpoints in /entities, /groups and /markets instead.

### Optional Parameters

#### name

- **type:** string
- **description:** Name of the stat

#### relatedEntityType

- **type:** string
- **description:** Used to get stats from different entity types in the supply chain of {uuid} E.G.costPrice of the VMs where the scope is a PM

#### filters

- **type:** array
- **description:** Filters to apply to the requested stats. Commodities supported values: key = {commodity\_key}, relation = sold|bought, virtualDisk = {uuid}. Cost Price supported values: cumulative = true|false, cloudService = {uuid}, target = {uuid}, CSP = {name}, category = {name}, businessUnit = {uuid}, costComponent = COMPUTE|IP|LICENSE|STORAGE, costComponentArtifact = {key}. Please note that the costPrice stat is deprecated and will EOL in 8.10.1. Use the dedicated 'cost' endpoints in /entities, /groups and /markets instead. Action stats supported values: actionTypes = {actionType}, actionModes: {action\_mode}, actionStates: {action\_state}, riskSubCategory: Performance Assurance|Efficiency Improvement|Prevention|Compliance, riskSeverity: UNKNOWN|NORMAL|MINOR|MAJOR|CRITICAL
- – **StatFilterApiDTO**

#### groupBy

- **type:** array



- **description:** Execute a groupBy on the values of the stats. Commodities supported values: key, relatedEntity, virtualDisk. Cost Price supported values: cloudService, target, CSP, category, businessUnit, costComponent, costComponentArtifact. Please note that the costPrice stat is deprecated and will EOL in 8.10.1. Use the dedicated 'cost' endpoints in /entities, /groups and /markets instead. Action stats supported values: actionTypes, actionModes, actionStates, risk, riskSubCategory, riskSeverity
- – **type:** string

#### historyType

- **type:** string
- **description:** Query the histUtilization table for these historyTypes
- **enum:** ['Percentile', 'Smoothed']

## StatCommoditySourceApiDTO

### Description

Model to describe commodity source for a single statistic, e.g. commodity source for VCPU (capacity and usage)

### Optional Parameters

#### capacityCommoditySource

- **type:** string
- **description:** Commodity source for capacity

#### usedCommoditySource

- **type:** string
- **description:** Commodity source for usage

## StatFilterApiDTO

### Description

Model to describe the filter applied to a Statistic

### Required Parameters

#### type

- **type:** string
- **description:** type of the filter, E.G: actionTypes, category, ...

### Optional Parameters

#### value

- **type:** string
- **description:** value of the filter

## StatHistUtilizationApiDTO

### Description

Model to describe a historical utilization statistic, e.g. the P95 vCPU or P95 vMEM of a VM over (up to) NN days observation period.

### Optional Parameters

#### type

- **type:** string
- **description:** Historical utilization type

#### usage

- **type:** number
- **format:** float
- **description:** Historical usage

#### capacity

- **type:** number
- **format:** float
- **description:** Historical capacity

#### resizeMaxScalingObservationPeriod

- **type:** number
- **format:** float
- **description:** Resize max scaling observation period. Affects on amount of data points considered for action generation process. Value in days.

#### resizeScalingAggressiveness

- **type:** number
- **format:** float
- **description:** Resize scaling aggressiveness. Describes how aggressively Turbonomic will resize in response of resource utilization. Value in percents. 100% least aggressive, 95% most aggressive.

## StatPeriodApiInputDTO

### Description

Model to describe the request for Statistics by a Time range

### Required Parameters

#### statistics

- **type:** array
- **description:** List of requested statistics, if empty all stats will be processed
- – [StatApiInputDTO](#)

### Optional Parameters

#### startDate

- **type:** string

- **description:** Start time of the period to filter, if empty the current stats will be processed

#### endDate

- **type:** string
- **description:** End time of the period to filter, if empty the current stats will be processed

## StatScopesApiInputDTO

### Description

Model to describe the request for Statistics in multiple Scopes

### Optional Parameters

#### scopes

- **type:** array
- **description:** List of uuid to use as scope
- – **type:** string

#### period

- **description:** List of requested statistics
- [StatPeriodApiInputDTO](#)

#### relatedType

- **type:** string
- **description:** Get the stats for the entity type related to the scopes

## StatSnapshotApiDTO

### Description

Model to describe the Statistics of a snapshot, contains a list of Statistics for a moment in time

### Required Parameters

#### statistics

- **type:** array
- **description:** List of statistics, e.g. Price Index, Commodities values, Cost price, Number of VMs in a Group, ...
- – [StatApiDTO](#)

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

### **date**

- **type:** string
- **description:** When the statistics were recorded

### **epoch**

- **type:** string
- **description:** An indicator of whether the snapshot represents historical, projected or plan data. Should be used in conjunction with the date field to understand the context of this stat snapshot.
- **enum:** ['HISTORICAL', 'CURRENT', 'PROJECTED', 'PLAN\_SOURCE', 'PLAN\_PROJECTED']

## **StatValueApiDTO**

### **Description**

Model to describe a single statistic, e.g. the CPU of a PM Group

### **Optional Parameters**

#### **max**

- **type:** number
- **format:** float
- **description:** Max value

#### **min**

- **type:** number
- **format:** float
- **description:** Min value

#### **avg**

- **type:** number
- **format:** float
- **description:** Average value

#### **total**

- **type:** number
- **format:** float
- **description:** Total value - sum of average values in a group

#### **totalMax**

- **type:** number
- **format:** float
- **description:** Total max value - sum of max values in a group

#### **totalMin**

- **type:** number
- **format:** float
- **description:** Total min value - sum of min values in a group

## StringSettingApiDTO

### Description

### Inherits From

This DTO inherits from [SettingApiDTO](#)

### Required Parameters

#### value

- **type:** string
- **description:** Value of the setting. Type indicated by the `valueObjectType` field. The default type is `String`

### Optional Parameters

#### defaultValue

- **type:** string
- **description:** Default value before any change
- **readOnly:** True

## SupplyChainStatsApiInputDTO

### Description

Model to describe a Filter for Entities

### Required Parameters

#### uuids

- **type:** array
- **description:** List of uuids used to build the supply chain
- – **type:** string
- **maxItems:** 2147483647
- **minItems:** 1

### Optional Parameters

#### types

- **type:** array
- **description:** List of Types to filter
- – **type:** string

#### states

- **type:** array
- **description:** List of States to filter
- – **type:** string

- **enum:** ['IDLE', 'RESOURCE\_ALLOCATION', 'LAUNCH', 'ACTIVE', 'RESOURCE\_RELEASE', 'SUSPEND', 'SUSPEND\_PENDING', 'TERMINATE\_PENDING', 'NOT\_MONITORED', 'MAINTENANCE', 'FAILOVER', 'UNKNOWN', 'EVACUATED', 'QUEUED']

#### **groupBy**

- **type:** array
- **description:** List of fields used to group the Stats
- – **type:** string
  - **enum:** ['entityType', 'state', 'severity', 'riskSubCategory', 'template', 'target', 'businessUnit', 'resourceGroup', 'nodePool']

#### **environmentType**

- **type:** string
- **description:** Filter the actions by Environment Type
- **enum:** ['ONPREM', 'CLOUD']

## **SupplychainApiDTO**

### **Description**

Model to describe the Supply Chain: the Environment is described as a chain between Producers and Consumers

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **seMap**

- **type:** object
- **description:** Model to describe the type of Entity associated with a SupplyChainEntry
- **additionalProperties:** {'\$ref': '#/definitions/SupplychainEntryDTO'}

## **SupplychainEntryDTO**

### **Description**

Model to describe a Supply Chain Entry

### **Optional Parameters**

#### **uuid**

- **type:** string

**displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**depth**

- **type:** integer
- **format:** int32
- **description:** Number to indicate the level of depth of the Entity type in the Supply Chain

**entitiesCount**

- **type:** integer
- **format:** int32
- **description:** Number of entity in this level of depth

**healthSummary**

- **type:** object
- **description:** Entities count by Severity
- **additionalProperties:** {'type': 'integer', 'format': 'int32'}

**stateSummary**

- **type:** object
- **description:** Entities count by State
- **additionalProperties:** {'type': 'integer', 'format': 'int32'}

**connectedProviderTypes**

- **type:** array
- **description:** List of provider types connected to this level of depth
- **uniqueItems:** True
- – **type:** string

**connectedConsumerTypes**

- **type:** array
- **description:** List of consumer types connected to this level of depth
- **uniqueItems:** True
- – **type:** string

**instances**

- **type:** object
- **description:** List of entities in this level of depth
- **additionalProperties:** {'\$ref': '#/definitions/ServiceEntityApiDTO'}

## TagApiDTO

**Description**

Tag information

## Optional Parameters

### key

- **type:** string
- **description:** Tag key

### values

- **type:** array
- **description:** Tag value
- – **type:** string

## TargetApiDTO

### Description

Model to describe a Target. Targets are used to discover the components to create a connected virtual environment

### Required Parameters

#### type

- **type:** string
- **description:** Probe type. Turbonomic ships with the following probe types: AppDynamics, Dynatrace, Datadog, Arista, AWS, Azure, Cisco APIC, Cisco HyperFlex, Cisco UCS Central, Cisco UCS Director, Cisco UCS Manager, CloudFoundry, CloudStack, Dell Compellent, EMC ScaleIO, EMC VMAX, EMC VNX, EMC VPLEX, EMC XtremIO, Hitachi Vantara, HP OneView, HPE 3PAR, Hyper-V, IBM PowerVM, Instana, JBoss, JVM, MExchange, MySQL, NetApp, NetFlow, NetScaler, Nutanix, OpenStack, Oracle, Pivotal Ops Manager, Pure, Red Hat Virtualization Manager, sFlow, SNMP, SoftLayer, SQLServer, Tetration, Tomcat, vCenter, vCloudDirector, VMM, WebLogic, WebSphere, WMI, XenServer,

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### category

- **type:** string
- **example:** Hypervisor
- **description:** Probe category.

#### uiCategory

- **type:** string
- **example:** Application Server, Billing, Cloud Management, Applications and Databases, Fabric and Network, Guest OS Processes, HYPERCONVERGED, HYPERVISOR, Load Balancer, Operations



Manager Appliance, ORCHESTRATOR, PaaS, Storage, Storage Browsing, WINDOWS APPLICATION. There are also categories Custom and UNKNOWN to be used for new probe development.

- **description:** Probe UI category.

#### identifyingFields

- **type:** array
- **description:** Array of field names used to differentiate probes of the same category. Most typically either the name/address field, or a combination of address and port.
- – **type:** string

#### inputFields

- **type:** array
- **description:** List of all InputFieldApiDTO objects for the target, analogous to the target addition form fields in the GUI.
- – **InputFieldApiDTO**

#### lastValidated

- **type:** string
- **description:** Date of the last validation - [DEPRECATED - Use: healthSummary.timeOfLastSuccessfulDiscovery]

#### status

- **type:** string
- **example:** Validated, VALIDATION FAILED, UNKNOWN
- **description:** Description of the status of the most recent Validation Operation - [DEPRECATED - Use: healthSummary.healthState]

#### derivedTargets

- **type:** array
- **description:** Targets derived (created by discovery of) from this target. Populated depending on the TargetRelationship parameter.
- – **TargetApiDTO**

#### patchedTargets

- **type:** array
- **description:** Targets patched by this target
- – **PatchedTargetDataApiDTO**

#### parentTargets

- **type:** array
- **description:** Targets that created this target. Populated depending on the TargetRelationship parameter.
- – **TargetApiDTO**

#### readonly

- **type:** boolean
- **description:** Whether the target cannot be changed through public APIs.

#### health

- **description:** [BETA]: A summary of the overall health of the target. Only returned if TargetDetailLevel is HEALTH or HEALTH\_DETAILS
- **TargetHealthApiDTO**

### lastTargetOperationStages

- **type:** array
- **description:** [BETA]: The status at each stage of the most recent validation or discovery, which ever was more recent. Only returned if TargetDetailLevel is HEALTH\_DETAILS
- – **TargetOperationStageApiDTO**

### lastEditTime

- **type:** string
- **example:** 2021-07-07T20:33:49Z
- **description:** [BETA]: The last time the target was edited in ISO-8601 Format (YYYY-MM-DDTHH:MM:SSZ). For example 2021-07-07T20:33:49Z

### lastEditUser

- **type:** string
- **description:** [BETA]: The username for last user who edited the target.

### healthSummary

- **description:** The health summary of the target.
- **TargetHealthSummaryApiDTO**

## TargetErrorDetailsApiDTO

### Description

Additional details related to a target error.

### Required Parameters

#### targetErrorDetailsClass

- **type:** string

## TargetHealthApiDTO

### Description

A DTO that describes the health of a single target.

### Required Parameters

#### healthCategory

- **type:** string
- **description:** The type of the health checks that are done.
- **enum:** ['TARGET', 'ACTION', 'COMPONENT']

#### healthClassDiscriminator

- **type:** string
- **example:** TargetHealthApiDTO
- **description:** Health type discriminator. Describes which subclass of BaseHealthApiDTO is returned.

#### rollupState

- **type:** string
- **example:** NORMAL

- **description:** The health state of the target including its derived targets.
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

#### targetName

- **type:** string
- **description:** Target display name.

#### uuid

- **type:** string
- **description:** UUID of the target.

### Optional Parameters

#### errorText

- **type:** string
- **description:** Error text (if there's any).

#### timeOfFirstFailure

- **type:** string
- **description:** Time of first failure of validation/discovery (if the target is in failed state; ISO-8601 formatted String).

#### timeOfLastSuccessfulDiscovery

- **type:** string
- **example:** 2021-07-07T20:33:49Z
- **description:** [BETA]: The last time a successful discovery finished on the target in ISO-8601 Format (YYYY-MM-DDTHH:MM:SSZ). For example 2021-07-07T20:33:49Z

#### targetErrorDetails

- **type:** array
- **description:** Error details related to a target. Supports multiple errors details.
- – [TargetErrorDetailsApiDTO](#)

#### targetStatusSubcategory

- **type:** string
- **enum:** ['DISCOVERY', 'VALIDATION', 'DUPLICATION', 'DELAYED\_DATA']

#### healthState

- **type:** string
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

## TargetHealthSummaryApiDTO

### Description

A DTO that describes target health.

### Required Parameters

#### healthState

- **type:** string
- **example:** NORMAL

- **description:** The target health state.
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

#### rollupState

- **type:** string
- **example:** NORMAL
- **description:** The health state of the target including its derived targets.
- **enum:** ['CRITICAL', 'MAJOR', 'MINOR', 'NORMAL']

### Optional Parameters

#### timeOfLastSuccessfulDiscovery

- **type:** string
- **example:** 2021-07-07T20:33:49Z
- **description:** The last time a successful discovery finished on the target in ISO-8601 Format (YYYY-MM-DDTHH:MM:SSZ). For example 2021-07-07T20:33:49Z

## TargetOperationStageApiDTO

### Description

Holds all information related to a stage of discovery or validation.

### Optional Parameters

#### description

- **type:** string
- **description:** The description of what this stage is doing (ideally in less that 70 characters). Example of description are "validating credentials" or "retrieving business applications".

#### status

- **description:** Holds all the information of this particular instance of the discovery or validation stage.
- [TargetOperationStageStatusApiDTO](#)

## TargetOperationStageStatusApiDTO

### Description

Holds all the information of this particular instance of the discovery or validation stage.

### Optional Parameters

#### state

- **type:** string
- **description:** The state of this stage at the end of the discovery or validation.
- **enum:** ['DID\_NOT\_RUN', 'SUCCESS', 'FAILURE']

#### summary

- **type:** string
- **description:** The short message (ideally 70 characters or less) explaining the stage status. Example of such messages are "invalid credentials" or "4 business application retrieved".

### fullExplanation

- **type:** string
- **description:** The full message explaining the stage status. This explanation may include the instructions or hints on how to address issue if there is one.

## TargetThirdPartyErrorDetailsApiDTO

### Description

### Inherits From

This DTO inherits from [TargetErrorDetailsApiDTO](#)

### Optional Parameters

#### targetErrorType

- **type:** string
- **description:** Target validation/discovery error type.
- **enum:** ['INTERNAL\_PROBE\_ERROR', 'CONNECTIVITY\_ERROR', 'UNAUTHENTICATED', 'TOKEN\_UNAVAILABLE', 'VERSION\_NOT\_SUPPORTED', 'DATA\_ACCESS\_ERROR', 'DUPLICATION', 'DELAYED\_DATA', 'THIRD\_PARTY\_FAILURE']

#### errorCode

- **type:** string
- **example:** 0x423232, 404, 400
- **description:** Error code related to the failure.

#### endPoint

- **type:** string
- **description:** End point which caused the third party api failure. This can be a complete URL or a method name while using the third party SDK.

## TelemetryDTO

### Description

Model to describe the status of telemetry

### Optional Parameters

#### telemetryTermsViewed

- **type:** boolean
- **description:** Controls when an administrator should be shown the telemetry opt in

#### telemetryEnabled

- **type:** boolean
- **description:** Controls whether analytics and telemetry are enabled

# TemplateApiDTO

## Description

Model to describe a Template: contains the resources used to Deploy a VM or to Add Workload/Supply in a Plan

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### price

- **type:** number
- **format:** float
- **description:** The estimated cost of the template. Used for Physical Machine and Storage templates where the hardware cost is known.

### model

- **type:** string
- **example:** Cisco UCS B200 M5 (Intel Xeon Gold 5115, 2.40 GHz), PowerEdge R940 (Intel Xeon Gold 6136, 3.00 GHz), etc.
- **description:** The model of the entire Physical Machine system.

### cpuModel

- **type:** string
- **example:** Intel Xeon Gold 6130, AMD Opteron 6166 HE, etc.
- **description:** The model of the CPU in the Physical Machine.

### vendor

- **type:** string
- **example:** Cisco, Microsoft, HP, Novell
- **description:** Vendor of the templated entity

### description

- **type:** string
- **description:** Description

### image

- **type:** string
- **description:** Container Profile image

### imageTag

- **type:** string
- **description:** (UNUSED) Container Profile image tag

### **cmdWithArgs**

- **type:** string
- **description:** (UNUSED) Container command with Arguments

### **dbEngine**

- **type:** string
- **example:** PostgreSQL, AuroraMySQL, SqlServer
- **description:** Database engine, used for Database templates

### **dbEdition**

- **type:** string
- **example:** Standard One, Enterprise, Premium\_P2
- **description:** Database edition, used for Database templates

### **computeResources**

- **type:** array
- **description:** ResourceApiDTO describing the compute resources. Max 1 per template.
- – **ResourceApiDTO**
- **maxItems:** 1
- **minItems:** 0

### **storageResources**

- **type:** array
- **description:** ResourceApiDTO describing the storage resources
- – **ResourceApiDTO**

### **networkResources**

- **type:** array
- **description:** ResourceApiDTO describing the network resources
- – **ResourceApiDTO**

### **infrastructureResources**

- **type:** array
- **description:** ResourceApiDTO describing the infrastructure resources. Max 1 per template
- – **ResourceApiDTO**
- **maxItems:** 1
- **minItems:** 0

### **deploymentProfile**

- **description:** Deployment profile associated with the template
- **DeploymentProfileApiDTO**

### **discovered**

- **type:** boolean
- **example:** False
- **description:** Indicates if the template is discovered or manually created
- **enum:** [True, False]

### **family**

- **type:** string

- **example:** standardDSv2Family, standardNCFamily, r5d, r3
- **description:** For cloud templates, the instance family.

#### **enableMatch**

- **type:** boolean
- **example:** False
- **description:** Add to Infrastructure Cost Policy. Infrastructure Cost policies group hardware devices according to their cost

## **TemplateApiInputDTO**

### **Description**

Model to describe a Template: contains the resources used to Deploy a VM or to Add Workload/Supply in a Plan

### **Optional Parameters**

#### **displayName**

- **type:** string
- **description:** Name of the template

#### **className**

- **type:** string
- **description:** className
- **enum:** ['VirtualMachine', 'PhysicalMachine', 'Storage', 'Container']

#### **price**

- **type:** number
- **format:** float
- **description:** Cost price

#### **model**

- **type:** string
- **description:** The Model of the entire Physical Machine system. For example: Cisco UCS B200 M5 (Intel Xeon Gold 5115, 2.40 GHz) vs. PowerEdge R940 (Intel Xeon Gold 6136, 3.00 GHz)

#### **cpuModel**

- **type:** string
- **description:** The model of the CPU in the Physical Machine. This is separate from the model of the entire Physical Machine. For example: Intel Xeon Gold 6130 vs. AMD Opteron 6166 HE

#### **vendor**

- **type:** string
- **description:** Vendor

#### **description**

- **type:** string
- **description:** Description

#### **image**

- **type:** string



- **description:** Profile image, used for Container templates

#### **imageTag**

- **type:** string
- **description:** Profile image tag, used for Container templates

#### **cmdWithArgs**

- **type:** string
- **description:** Command with Arguments, used for Container templates

#### **computeResources**

- **type:** array
- **description:** Compute resources: Number of CPU, CPU speed, Memory size, ...
- – **ResourceApiDTO**
- **maxItems:** 1
- **minItems:** 0

#### **storageResources**

- **type:** array
- **description:** Storage resources: Disk I/O, Disk Size, Percentage of Disk consumed
- – **ResourceApiDTO**

#### **networkResources**

- **type:** array
- **description:** Network resources
- – **ResourceApiDTO**

#### **infrastructureResources**

- **type:** array
- **description:** Infrastructure resources: Power, Size, Cooling
- – **ResourceApiDTO**
- **maxItems:** 1
- **minItems:** 0

#### **deploymentProfileId**

- **type:** string
- **description:** Ids of the Deployment Profiles associated with this Template

#### **enableMatch**

- **type:** boolean
- **description:** Add to Infrastructure Cost Policy. Infrastructure Cost policies group hardware devices according to their cost

## **TemplatePriceAdjustmentDTO**

### **Description**

Price adjustment information for a template within a business unit and specified pricing model.

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### environmentType

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

### discount

- **type:** number
- **format:** float
- **description:** Entity discount percentage

### priceAdjustment

- **description:** Entity price adjustment percentage
- **PriceAdjustmentDTO**

### family

- **type:** string
- **description:** Family of a template, group of similar purpose templates. Available only for virtual machine templates.

### pricesPerDatacenter

- **type:** array
- **description:** Template base rates per data center, provided for information purpose.
- – **EntityPriceDTO**

## TextConditionApiDTO

### Description

### Inherits From

This DTO inherits from [ConditionApiDTO](#)

### Required Parameters

#### value

- **type:** string
- **description:** A regular expression to be matched against the value of the field.
- **readOnly:** True

## TextFieldValueApiDTO

### Description

### Inherits From

This DTO inherits from [FieldValueApiDTO](#)

### Required Parameters

#### value

- **type:** string
- **description:** The value.
- **readOnly:** True

## TimeBasedTopologyChangesApiDTO

### Description

Topology changes based on a timeframe in the scenario

### Optional Parameters

#### includeReservation

- **type:** boolean
- **description:** Add the workload that is currently reserved

#### reservationCount

- **type:** integer
- **format:** int32
- **description:** # of currently reserved VMs
- **readOnly:** True

#### addHistoryVMs

- **type:** boolean
- **description:** Should historical vmNetGrowth be considered

#### vmNetGrowth

- **type:** integer
- **format:** int32
- **description:** Number of workloads added during the last month
- **readOnly:** True

## TimeSpanApiDTO

### Description

Schedule Time span information

## Required Parameters

### begins

- **type:** string
- **description:** Start time of the time span
- **pattern:** ^([0-1]?[0-9]|2[0-3]):[0-5][0-9]\$

### ends

- **type:** string
- **description:** End time of the time span
- **pattern:** ^([0-1]?[0-9]|2[0-3]):[0-5][0-9]\$

### policy

- **description:** policy to be applied during this time span
- [ScheduleItemApiDTO](#)

## TimeSpanGroupApiDTO

### Description

Base model to describe a TimeSpanGroup.

### Required Parameters

#### type

- **type:** string
- **example:** WeekDayTimeSpansApiDTO
- **description:** TimeSpan group type discriminator. Describes which subclass of TimeSpanGroupApiDTO is returned.

## TopologyChangesApiDTO

### Description

Topology changes in the scenario

### Optional Parameters

#### addList

- **type:** array
- **description:** List of add changes in the Topology
- – [AddObjectApiDTO](#)

#### removeList

- **type:** array
- **description:** List of remove changes in the Topology
- – [RemoveObjectApiDTO](#)

#### replaceList

- **type:** array
- **description:** List of replace changes in the Topology
- – [ReplaceObjectApiDTO](#)

### **migrateList**

- **type:** array
- **description:** List of migrate changes in the Topology
- – **MigrateObjectApiDTO**

### **relievePressureList**

- **type:** array
- **description:** List of relieve pressure changes in the Topology
- – **RelievePressureObjectApiDTO**

## **TopologyDataDefinitionApiDTO**

### **Description**

Model to describe a user-defined topology definition.

### **Required Parameters**

#### **entityDefinitionData**

- **description:** Entity definition data
- **IEntityDefinitionData**

#### **entityType**

- **type:** string
- **description:** The type of service entity being defined
- **enum:** ['BusinessApplication', 'BusinessTransaction', 'Service']

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### **className**

- **type:** string

#### **contextBased**

- **type:** boolean
- **description:** Defines if this Definition is Context-Based.

## **UnplacementDetailsApiDTO**

### **Description**

Additional information provided for entities in a plan that are unplaced

## Required Parameters

### reasons

- **type:** array
- **description:** Detailed information about why the entity could not be placed
- – [UnplacementReasonApiDTO](#)

## Optional Parameters

### placedOn

- **type:** array
- **description:** For Unplaced entities in a Plan Market, the providers that this entity was able to place on
- – [BaseApiDTO](#)

## UnplacementReasonApiDTO

### Description

Details to describe why an entity could not be placed

### Optional Parameters

#### failedResources

- **type:** array
- **description:** A list of resources that caused the entity not to be placed
- – [FailedResourceApiDTO](#)

#### resourceOwner

- **description:** The entity owning the resources that caused unplacement, if different than this entity. For example, a volume that caused a VM to be unplaced.
- [BaseApiDTO](#)

#### providerType

- **type:** string
- **description:** The entity type that sells the commodities that caused unplacement.
- **enum:** ['Application', 'ApplicationComponent', 'ApplicationServer', 'AvailabilityZone', 'BusinessAccount', 'BusinessApplication', 'BusinessTransaction', 'BusinessUser', 'Chassis', 'CloudService', 'ComputeTier', 'Container', 'ContainerPod', 'ContainerSpec', 'Database', 'DatabaseServer', 'DatabaseServerTier', 'DatabaseTier', 'DataCenter', 'DesktopPool', 'DiskArray', 'DPod', 'HCIPhysicalMachine', 'HypervisorServer', 'Internet', 'IOModule', 'LoadBalancer', 'LogicalPool', 'Namespace', 'Network', 'PhysicalMachine', 'ProcessorPool', 'Region', 'ReservedInstance', 'Service', 'ServiceProvider', 'Storage', 'StorageController', 'StorageTier', 'Switch', 'Unknown', 'ViewPod', 'VirtualApplication', 'VirtualDataCenter', 'VirtualMachine', 'VirtualVolume', 'VPod', 'WorkloadController', 'ContainerPlatformCluster', 'CloudCommitment', 'ApplicationComponentSpec', 'VirtualMachineSpec', 'DocumentCollection']

#### closestSeller

- **description:** The seller that came closest to satisfying the commodities. This information is not always available.
- [BaseApiDTO](#)

## placementProblem

- **type:** string
- **description:** Categorizes the reason which caused unplacement
- **enum:** ['COSTS\_NOT\_FOUND', 'NOT\_CONTROLLABLE', 'UNSATISFIED\_COMMODITIES']

## UserApiDTO

### Description

Model to describe an User

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### username

- **type:** string
- **description:** Username

#### password

- **type:** string
- **description:** Encoded password

#### roles

- **type:** array
- **description:** Roles assigned to the user.
- – [RoleApiDTO](#)

#### loginProvider

- **type:** string
- **description:** Provider for the login
- **enum:** ['Local', 'LDAP']

#### type

- **type:** string
- **description:** type
- **enum:** ['DedicatedCustomer', 'SharedCustomer']

#### scope

- **type:** array
- **description:** List of scopes that the User is allowed to see
- – [GroupApiDTO](#)

### features

- **type:** array
- **description:** List of features that the user is allowed to see
- – **type:** string

### authToken

- **type:** string
- **description:** The secured user token

### showSharedUserSC

- **type:** boolean

## UtilizationApiDTO

### Description

Model to describe a utilization change in the scenario

### Optional Parameters

#### projectionDay

- **type:** integer
- **format:** int32
- **description:** Projection day, indicate when to apply the change

#### target

- **description:** Target of the change, entity or group
- **BaseApiDTO**

#### percentage

- **type:** integer
- **format:** int32
- **description:** Utilization percentage change

## VMEntityAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

#### os

- **type:** string
- **description:** The operating system type. E.G. 'Linux', 'Windows' and 'Unknown' if not found.

#### ip

- **type:** array
- **description:** The IP address. Can be IPv4 or IPv6



- – **type:** string

#### **connectedNetworks**

- **type:** array
- **description:** The connected Networks
- – **BaseApiDTO**

#### **numVCPUs**

- **type:** integer
- **format:** int32
- **description:** Number of vCPUs

#### **coresPerSocketRatio**

- **type:** integer
- **format:** int32
- **description:** Cores per socket ratio

#### **processorPoolName**

- **type:** string
- **description:** Processor Pool

#### **hasDedicatedProcessors**

- **type:** boolean
- **example:** True
- **description:** Has Dedicated Processors

#### **processorCompatibilityMode**

- **type:** string
- **example:** POWER8
- **description:** Processor Compatibility Mode

#### **activeMemoryExpansionEnabled**

- **type:** boolean
- **example:** True
- **description:** Active Memory Expansion Enabled

#### **resourceMonitoringControlState**

- **type:** string
- **example:** active
- **description:** Resource Monitoring Control (RMC) State

#### **sharingMode**

- **type:** string
- **example:** uncapped
- **description:** Sharing Mode

#### **dedicatedSharingMode**

- **type:** string
- **example:** share\_idle\_pros
- **description:** Dedicated Sharing Mode

### **nonDisruptiveMaxPU**

- **type:** number
- **format:** double
- **description:** Non-disruptive Processing Units max resize constraint

### **nonDisruptiveMinPU**

- **type:** number
- **format:** double
- **description:** Non-disruptive Processing Units min resize constraint

### **ebsOptimized**

- **type:** boolean
- **description:** EBS Optimized

### **sessions**

- **type:** array
- **description:** Information about sessions established by a BusinessUser to this VirtualMachine.
- – **BusinessUserSessionApiDTO**

### **vendorToolsVersion**

- **type:** string
- **description:** Vendor Tools Version

### **resourceId**

- **type:** string
- **description:** Resource identifier of the virtual machine. For example, the resource id in Azure, the ARN in AWS, or the full resource name in GCP.

### **creationTimeStamp**

- **type:** integer
- **format:** int64
- **description:** Timestamp in milliseconds for the creation of the virtual machine.

### **vgpuProfile**

- **type:** string
- **description:** vGPU profile used by virtual machine.

### **ahublicense**

- **type:** boolean

## **VirtualDiskApiDTO**

### **Description**

Model to describe a Virtual Disk

### **Optional Parameters**

#### **uuid**

- **type:** string

**displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**tier**

- **type:** string
- **description:** The tier of the Virtual Disk

**actions**

- **type:** array
- **description:** The related action: Move or Delete
- – [ActionApiDTO](#)

**stats**

- **type:** array
- **description:** The related statistics
- – [StatApiDTO](#)

**attachedVirtualMachine**

- **description:** The Virtual Machine if the disk is attached
- [BaseApiDTO](#)

**provider**

- **description:** The Storage provider
- [BaseApiDTO](#)

**dataCenter**

- **description:** The DataCenter where the VirtualDisk resides
- [BaseApiDTO](#)

**environmentType**

- **type:** string
- **description:** Environment type
- **enum:** ['CLOUD', 'ONPREM', 'HYBRID']

**lastModified**

- **type:** integer
- **format:** int64
- **description:** Last Modified

**businessAccount**

- **description:** Business Account
- [BaseApiDTO](#)

**resourceGroup**

- **description:** Resource Group
- [BaseApiDTO](#)

**creationTime**

- **type:** string
- **description:** Creation time

**skuName**

- **type:** string
- **description:** Sku name

**snapshotId**

- **type:** string
- **description:** Snapshot id

**encryption**

- **type:** string
- **description:** Encryption

**attachmentState**

- **type:** string
- **description:** Attachment state
- **enum:** ['AVAILABLE', 'IN\_USE', 'UNKNOWN']

**name**

- **type:** string
- **description:** Name

**ephemeral**

- **type:** string
- **description:** Ephemeral

**lastAttachedVm**

- **type:** string
- **description:** Name of the VM this volume was last attached to.

**numDaysUnattached**

- **type:** string
- **description:** Number of days ago this volume was last attached.

**hourlyBilledOps**

- **type:** number
- **format:** double
- **description:** Average number of billed operations per hour.

**creationTimeStamp**

- **type:** integer
- **format:** int64
- **description:** Timestamp in milliseconds for the creation of the virtual disk.

**attachedVMStorageCompatibility**

- **type:** string
- **description:** Storage compatibility for virtualDisks's attached VM(s). The compatibility will affect scale analysis for the volume. If an attached VM is not PREMIUM compatible, the volume will never scale to a premium StorageTier

- **enum:** ['PREMIUM', 'STANDARD']

#### **storageUsageType**

- **type:** string
- **description:** Usage type for the volume, i.e., what the volume is used for.
- **enum:** ['SITE\_RECOVERY', 'BACKUP', 'STANDARD']

#### **resourceId**

- **type:** string
- **description:** Resource identifier of the volume. For example, the resource id in Azure, the ARN in AWS, or the full resource name in GCP.

## **VirtualDisksAspectApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [EntityAspect](#)

### **Optional Parameters**

#### **virtualDisks**

- **type:** array
- **description:** List of Virtual Disks
- – [VirtualDiskApiDTO](#)

## **VirtualVolumeEntityAspectApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [EntityAspect](#)

### **Optional Parameters**

#### **files**

- **type:** array
- **description:** Files comprised in the volume
- – **type:** string

## **WebhookApiDTO**

### **Description**

### **Inherits From**

This DTO inherits from [WorkflowAspect](#)

## Required Parameters

### method

- **type:** string
- **description:** The http method used to make the request.
- **enum:** ['GET', 'POST', 'PUT', 'DELETE', 'PATCH']

### url

- **type:** string
- **description:** The URL that HTTP request is made to.

## Optional Parameters

### template

- **type:** string
- **description:** The template for the body of request.

### authenticationMethod

- **type:** string
- **description:** The authentication method to use for the request.
- **enum:** ['NONE', 'BASIC', 'OAUTH']

### username

- **type:** string
- **description:** The username for the authenticated request.

### password

- **type:** string
- **description:** The password for the authenticated request.

### trustSelfSignedCertificates

- **type:** boolean
- **description:** If true, self-signed certificates will be trusted when using HTTPS connections. Defaults to 'false'.

### headers

- **type:** array
- **description:** The request headers.
- – **RequestHeader**

### oauthData

- **OAuthDataApiDTO**

## WeekDayTimeSpansApiDTO

### Description

### Inherits From

This DTO inherits from [TimeSpanGroupApiDTO](#)

## Optional Parameters

### sunday

- **type:** array
- **description:** Schedule time spans information for sunday
- – [TimeSpanApiDTO](#)

### monday

- **type:** array
- **description:** Schedule time spans information for monday
- – [TimeSpanApiDTO](#)

### tuesday

- **type:** array
- **description:** Schedule time spans information for tuesday
- – [TimeSpanApiDTO](#)

### wednesday

- **type:** array
- **description:** Schedule time spans information for wednesday
- – [TimeSpanApiDTO](#)

### thursday

- **type:** array
- **description:** Schedule time spans information for thursday
- – [TimeSpanApiDTO](#)

### friday

- **type:** array
- **description:** Schedule time spans information for friday
- – [TimeSpanApiDTO](#)

### saturday

- **type:** array
- **description:** Schedule time spans information for saturday
- – [TimeSpanApiDTO](#)

## WhereApiDTO

### Description

Filter for entity or group queries.

### Optional Parameters

#### relatedToScopes

- **type:** array
- **description:** Restrict to entities related to these scopes.
- **readOnly:** True
- – **type:** integer
- **format:** int64

### conditions

- **type:** array
- **description:** Restrict to entities satisfying these conditions.
- **readOnly:** True
- – ConditionApiDTO

## WidgetApiDTO

### Description

Basic information of an entity

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### type

- **type:** string

#### scope

- BaseApiDTO

#### startPeriod

- **type:** string

#### endPeriod

- **type:** string

#### row

- **type:** integer
- **format:** int32

#### column

- **type:** integer
- **format:** int32

#### sizeRows

- **type:** integer
- **format:** int32

#### sizeColumns

- **type:** integer
- **format:** int32



**widgetsetId**

- **type:** integer
- **format:** int64

**widgetElements**

- **type:** array
- – [WidgetElementApiDTO](#)

**widgetSource**

- **type:** string
- **enum:** ['UI\_DEFAULT', 'USER']

## WidgetElementApiDTO

**Description**

Basic information of an entity

**Optional Parameters****uuid**

- **type:** string

**displayName**

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**row**

- **type:** integer
- **format:** int32

**column**

- **type:** integer
- **format:** int32

**type**

- **type:** string

**widgetId**

- **type:** integer
- **format:** int64

**properties**

- **type:** object
- **additionalProperties:** {'type': 'string'}

**stats**

- **type:** array
- – [StatApiInputDTO](#)

## datasets

- **type:** array
- – **type:** string

## WidgetsetApiDTO

### Description

Basic information of an entity

### Optional Parameters

#### uuid

- **type:** string

#### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

#### className

- **type:** string

#### scope

- **type:** string

#### startPeriod

- **type:** string

#### endPeriod

- **type:** string

#### widgets

- **type:** array
- – WidgetApiDTO

#### category

- **type:** string

#### scopeType

- **type:** string

#### username

- **type:** string

#### isSharedWithAllUsers

- **type:** boolean

## WorkflowApiDTO

### Description

Model to describe a Workflow

## Required Parameters

### discoveredBy

- **description:** Target that discovered this workflow
- **TargetApiDTO**

### type

- **type:** string
- **description:** The type of this workflow
- **enum:** ['UCSD', 'ACTION\_SCRIPT', 'SERVICENOW', 'ACTIONSTREAM\_KAFKA', 'WEBHOOK']

## Optional Parameters

### uuid

- **type:** string

### displayName

- **type:** string
- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

### className

- **type:** string

### description

- **type:** string
- **description:** Description of the workflow

### entityType

- **type:** string
- **description:** Type of the entity associated

### actionType

- **type:** string
- **description:** Type of the action associated

### parameters

- **type:** array
- **description:** Input parameters for the workflow
- – **InputFieldApiDTO**

### scriptPath

- **type:** string
- **description:** Full path to workflow script (e.g. path to an ActionScript executable)

### actionPhase

- **type:** string
- **description:** Action Phase to which workflow applies

### timeLimitSeconds

- **type:** integer
- **format:** int64
- **description:** Time limit for workflow execution, in seconds

## **typeSpecificDetails**

- **description:** The information about this workflow that is specific to its type.
- **WorkflowAspect**

## **WorkflowAspect**

### **Description**

Parent model to describe a type-specific workflow details

### **Required Parameters**

#### **type**

- **type:** string

## **WorkflowOperationRequestApiDTO**

### **Description**

The details of the requested operation on the workflow.

### **Required Parameters**

#### **actionId**

- **type:** integer
- **format:** int64
- **description:** The ID of the action that we perform the workflow operation on

#### **operation**

- **type:** string
- **description:** The operation that is being performed on the workflow
- **enum:** ['TEST']

## **WorkflowOperationResponseApiDTO**

### **Description**

Result of a workflow operation.

### **Required Parameters**

#### **succeeded**

- **type:** boolean
- **description:** True if the workflow operation has succeeded.

### **Optional Parameters**

#### **uuid**

- **type:** string

#### **displayName**

- **type:** string

- **description:** A user readable name of the api object. Cannot be blank. The UTF-8 encoding must be at most 255 bytes.

**className**

- **type:** string

**details**

- **type:** string
- **description:** Details of workflow operation result.

## WorkloadControllerAspectApiDTO

### Description

### Inherits From

This DTO inherits from [EntityAspect](#)

### Optional Parameters

**controllerType**

- **type:** string
- **description:** Information about workload controller type.
- **enum:** ['CRON\_JOB\_INFO', 'CUSTOM\_CONTROLLER\_INFO', 'DAEMON\_SET\_INFO', 'DEPLOYMENT\_INFO', 'JOB\_INFO', 'REPLICA\_SET\_INFO', 'REPLICATION\_CONTROLLER\_INFO', 'STATEFUL\_SET\_INFO']

**customControllerType**

- **type:** string
- **description:** Information about custom controller type. Only set when the controllerType is CUSTOM\_CONTROLLER\_INFO.

**controllerReplicaCount**

- **type:** integer
- **format:** int32
- **description:** The number of active replicas controlled by the workload controller.

## Reference: Entity Statistics

---

Turbonomic monitors the following statistics for entities in your environment:

### ApplicationComponent Statistics

Heap

Produces

RemainingGcCapacity

ResponseTime

Threads

Transaction

VCPU

VMem

## BusinessApplication Statistics

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

### ResponseTime

Response time, measured in ms.

### Transaction

Transactions per second

## BusinessTransaction Statistics

### Produces

### ResponseTime

### Transaction

## BusinessUser Statistics

### ActiveSessions

### ImageCPU

### ImageMem

### ImageStorage

### PoolCPU

### PoolMem

### PoolStorage

### Produces

### TotalSessions

## Chassis Statistics

### Cooling

Allocated cooling indicates the highest acceptable running temperature for a physical device, such as a chassis in a compute fabric.

### Power

A measure of the power that is consumed by a physical device.

### priceIndex

The degree of risk for the chassis.

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

### Space

When known, the amount of physical hardware that can be used in conjunction with this entity. For example, the number of spaces in a server rack.

## ContainerPlatformCluster Statistics

### NumberConsumers

**Produces**  
**VCPU**  
**VCPURequest**  
**VMem**  
**VMemRequest**

## **ContainerPod Statistics**

**NumberConsumers**  
**Produces**  
**StorageAmount**  
**VCPU**  
**VCPULimitQuota**  
**VCPURequest**  
**VCPURequestQuota**  
**VMem**  
**VMemLimitQuota**  
**VMemRequest**  
**VMemRequestQuota**  
**VStorage**

## **ContainerSpec Statistics**

**Produces**  
**VCPU**  
**VCPURequest**  
**VCPUThrottling**  
**VMem**  
**VMemRequest**

## **Container Statistics**

**Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

**VCPU**

The percentage utilization of the VCPU allocated for the hosting entity.

**VCPULimitQuota**

**VCPURequest**

**VCPURequestQuota**

**VCPUThrottling**

**VMem**

The percentage utilization of the VMem allocated to the hosting VM.

**VMemLimitQuota**

**VMemRequest**

**VMemRequestQuota**

## DataCenter Statistics

### Ballooning

Ballooning capacity, measured in KBytes. This capacity is the greater of 65% of the VMem configured for all powered-on VMs that the PM hosts, or the physical memory capacity

### Cooling

Allocated cooling indicates the highest acceptable running temperature for a physical device, such as a chassis in a compute fabric.

### CPU

Host CPU capacity, measured in MHz. This shows what percentage of CPU cycles are devoted to processing instructions.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

### CPUAllocation

DESCRIPTION

### CPUProvisioned

The aggregate Overprovisioned CPU capacity and utilization on the host, in KB. Capacity is a function of the overprovisioning percentage specified for the host.

### HostLunAccess

### HotStorage

### IOThroughput

The data rate through the PM's IO adapters, measured in Kbytes per second.

### Mem

The aggregate percentage of the PM's memory that is reserved or in use, measured in Kbytes.

### MemAllocation

DESCRIPTION

### MemProvisioned

Overprovisioned MEM capacity and utilization on the host, in KB. Capacity is a function of the overprovisioning percentage specified for the host.



**NetThroughput**

The aggregate data rate through the PM's network adapters

**numCPUs**

Host CPU capacity, measured in MHz. This shows what percentage of CPU cycles are devoted to processing instructions

**numHosts**

The number of hosts in the Datacenter

**numSockets**

The number of sockets in the Datacenter

**Power**

A measure of the power that is consumed by a physical device.

**priceIndex**

The degree of risk for the datacenter

**Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

**Q16VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q1VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q2VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q32VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q4VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q64VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

**Q8VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## Space

When known, the amount of physical hardware that can be used in conjunction with this entity. For example, the number of spaces in a server rack.

## StorageAccess

IOPS capacity and utilization

## StorageLatency

## Swapping

DESCRIPTION

## DatabaseServer Statistics

### Connection

### DBCACHEHitRate

### DBMem

### priceIndex

### Produces

### ResponseTime

### Transaction

### TransactionLog

### VCPU

### VMem

### VStorage

## Database Statistics

### costPrice

DESCRIPTION

### DTU

### LicenseAccess

### priceIndex

The degree of risk for the database

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

### StorageAmount

## DesktopPool Statistics

### ActiveSessions

### CPUAllocation

### ImageCPU

### ImageMem

### ImageStorage

### MemAllocation

### PoolCPU

**PoolMem**  
**PoolStorage**  
**priceIndex**  
**Produces**  
**TotalSessions**

## DiskArray Statistics

**Extent**  
**priceIndex**  
**Produces**

## IOModule Statistics

### Cooling

Allocated cooling indicates the highest acceptable running temperature for a physical device, such as a chassis in a compute fabric.

### NetThroughput

The data rate through the entity's network adapters, measured in Kbytes/second

### PortChannel

DESCRIPTION

### Power

A measure of the power that is consumed by a physical device.

### priceIndex

The degree of risk for the IO Module

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

## LogicalPool Statistics

**DiskArrayAccess**  
**Extent**  
**priceIndex**  
**Produces**  
**StorageAccess**  
**StorageAmount**  
**StorageLatency**  
**StorageProvisioned**

## Namespace Statistics

**priceIndex**  
**Produces**  
**VCPU**  
**VCPULimitQuota**

**VCPURequest**  
**VCPURequestQuota**  
**VMem**  
**VMemLimitQuota**  
**VMemRequest**  
**VMemRequestQuota**

## **PhysicalMachine Statistics**

### **Ballooning**

Ballooning capacity, measured in KBytes. This capacity is the greater of 65% of the VMem configured for all powered-on VMs that the PM hosts, or the physical memory capacity.

### **Cooling**

#### **CPU**

CPU capacity, CPU reserved, and CPU utilized, in MHz.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

#### **CPUAllocation**

CPU capacity, as allocated to the host, in MHz.

#### **CPUProvisioned**

Overprovisioned CPU capacity and utilization on the host, in MHz. Capacity is a function of the overprovisioning percentage specified for the host.

#### **HostLunAccess**

#### **IOThroughput**

Capacity and utilization of data rate through the host's IO adapter, measured in Kbits/sec.

#### **Mem**

The percentage of the PM's memory that is reserved or in use, measured in Kbytes.

#### **MemAllocation**

Allocation and utilization of host memory, measured in KB.

#### **MemProvisioned**

Overprovisioned MEM capacity and utilization on the host, in KB. Capacity is a function of the overprovisioning percentage specified for the host.

## **NetThroughput**

Capacity and utilization of data rate through the host's Network adapter, measured in Kbits/sec.

## **numCPUs**

The number of CPUs on the host.

## **numSockets**

The number of sockets discovered on this host.

## **Power**

### **priceIndex**

The degree of risk for the host.

## **Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

## **Q16VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Q1VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Q2VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Q32VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Q4VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Q64VCPU**

## **Q8VCPU**

Ready queue capacity and wait time, measured in ms. Turbonomic monitors 1-CPU, 2-CPU, 4-CPU, up to 32-CPU ready queues.

## **Space**

### **StorageAccess**

### **StorageLatency**

### **Swapping**

The capacity, reserved capacity, and utilization for rate of memory swapping to disk, in bits per second.

## **Service Statistics**

### **NumberReplicas**

**priceIndex**

**Produces**

## **StorageController Statistics**

**priceIndex**

The degree of risk for the disk array.

**Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

**StorageAmount**

The storage capacity, reserved storage, and utilized storage for this entity's storage in MB

## **StorageTier Statistics**

**StorageAccess**

IOPS capacity and utilization

**StorageAmount**

The storage capacity, reserved storage, and utilized storage for this entity in MB

**StorageClusterCommodity**

**StorageLatency**

The capacity, reserved amount, and utilized amount of latency for storage transactions, in msec.

**StorageProvisioned**

Overprovisioned storage capacity and utilization on the entity, in MB. Capacity is a function of the overprovisioning percentage specified for the entity.

## **Storage Statistics**

**Extent**

**priceIndex**

The degree of risk for the storage entity.

**Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

**StorageAccess**

IOPS capacity and utilization

**StorageAmount**

The storage capacity, reserved storage, and utilized storage for this entity in MB

**StorageLatency**

The capacity, reserved amount, and utilized amount of latency for storage transactions, in msec.

**StorageProvisioned**

Overprovisioned storage capacity and utilization on the entity, in MB. Capacity is a function of the overprovisioning percentage specified for the entity.

## Switch Statistics

### NetThroughput

The data rate through the entity's network adapters, measured in Kbytes/second.

### PortChannel

The data rate through the port channel, measured in Kbytes/second.

### priceIndex

The degree of risk for the Switch.

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

## ViewPod Statistics

### ActiveSessions

### priceIndex

### Produces

### TotalSessions

## VirtualDataCenter Statistics

### CPUAllocation

Host CPU capacity, measured in MHz. This shows what percentage of CPU cycles are devoted to processing instructions

### MemAllocation

DESCRIPTION

### priceIndex

The degree of risk for the virtual datacenter

### Produces

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

## VirtualMachine Statistics

### Ballooning

Ballooning capacity, measured in KBytes. This capacity is the greater of 65% of the VMem configured for all powered-on VMs that the PM hosts, or the physical memory capacity

### CPU

Host CPU capacity, measured in MHz. This shows what percentage of CPU cycles are devoted to processing instructions

The percentage utilization of the VCPU allocated for the hosting VM.

### CPUAllocation

DESCRIPTION

**CPUProvisioned**

Overprovisioned CPU capacity and utilization on the entity, in Mhz. Capacity is a function of the overprovisioning percentage specified for the entity.

**IOThroughput**

DESCRIPTION

**Mem**

The percentage of the PM's memory that is reserved or in use by the VM, measured in Kbytes.

The percentage utilization of the VMem allocated by the host.

**MemAllocation**

DESCRIPTION

**MemProvisioned**

Overprovisioned Mem capacity and utilization on the entity, in Kbytes. Capacity is a function of the overprovisioning percentage specified for the entity.

**NetThroughput**

The data rate through the entity's network adapters, measured in Kbytes/second

**numVCPUs**

DESCRIPTION

**priceIndex**

The degree of risk for the Virtual Machine

**Produces**

**Not Implemented:** The number of entities connected to this provider, used in calculating pricing.

**StorageAccess**

IOPS capacity and utilization

**StorageAmount**

The storage capacity, reserved storage, and utilized storage for this entity's storage in MB

**StorageLatency**

DESCRIPTION

**StorageProvisioned**

Overprovisioned storage capacity and utilization on the entity, in MB. Capacity is a function of the overprovisioning percentage specified for the entity.

**Swapping**

DESCRIPTION

**VCPU**

The percentage utilization of the VCPU allocated for the hosting VM.

**VMem**

The percentage utilization of the VMem allocated by the host.



## VStorage

Virtual storage allocated to a VM, measured in Kbytes.

## VirtualVolume Statistics

**costPrice**

**IOThroughput**

**Produces**

**StorageAccess**

**StorageAmount**

**StorageLatency**

**StorageProvisioned**

## WorkloadController Statistics

**priceIndex**

**Produces**

**VCPULimitQuota**

**VCPURequestQuota**

**VMemLimitQuota**

**VMemRequestQuota**

## Reference: API Change History

---

Each major release of Turbonomic includes fixes issues and improvements to the API. This section lists these changes for each major release.

### Note:

Deprecations are implemented in point releases of Turbonomic. For example, if API features were deprecated in 8.6.3, you would see those deprecations linked here in the entry for 8.7.0, along with the list of all other fixes and improvements made in the 8.6.x point releases. For the latest deprecation history, see [API Deprecation History](#).

### 8.9.0 API Fixes and Improvements

#### • Improvement:

API users can now search for entities by their tag as follows: `GET /tags/entities?key=<tagKey>`.

API users can now search for entities by their tag by calling `GET /tags/entities?key=<tagKey>`, which includes entities with a slash (/) within a tag string.

#### Note:

The existing API call `GET /tags/<tagKey>/entites` will be permanently deleted in a future release. Until its removal, you can use both of these methods.

#### • Improvement:

API users can now filter and sort targets.

API users can now filter and sort the list of targets to restrict their view to only their relevant targets.

The following parameters were added to the list of targets request:

- target type (AWS, AWS\_BILLING, ACTIONSCRIPT, ACTIONSTREAM\_KAFKA, APPDYNAMICS, APPINSIGHTS, AZURE\_BILLING, AZURE\_EA, AZURE\_SERVICE\_PRINCIPAL,

CISCO\_HYPERFLEX, CISCO\_UCS\_MANAGER, DATADOG, DYNATRACE, EMC\_SCALEIO, EMC\_VMAX, FLEXERA, GCP\_BILLING, GCP\_SERVICE\_ACCOUNT, HP\_ONEVIEW, HPE\_3PAR, HYPER\_V, INSTANA, JBOSS, JVM, KUBERNETES, MYSQL, NETAPP, NEWRELIC, NUTANIX, ORACLE, PURE, SNMP, SQLSERVER, SERVICENOW, TOMCAT, VMM, VMWARE\_HORIZON, WMI, WEBLOGIC, WEBSPHHERE, VCENTER)

- validation status (FAILED, IN\_PROGRESS, SUCCESS)
- health state (CRITICAL, MAJOR, MINOR, NORMAL)

The following parameters for sorting the list of targets were added: `target` `type` and `health` `state`.

- **Fixed Issue:**

Kubeturbo is using a deprecated API for the `APIRequestCount` query.

Kubeturbo is using a deprecated API for the `APIRequestCount` query.

## 8.8.0 API Fixes and Improvements

- **Fixed Issue:**

For API developers, when executing requests on the Search endpoint, the API does not paginate the returned result.

For API developers, when executing requests on the Search endpoint, the API does not paginate the returned result.

## 8.7.0 API Fixes and Improvements

- **Improvement:**

The API now supports reservations with cluster affinity.

When creating reservations, you can now specify affinity rules for the reserved VMs. You can ensure all the VMs in a reservation deploy to the same cluster. If you do not specify affinity, the VMs deploy to any clusters in the scope. This is the same as earlier behavior for reservations.

For more information, see [Affinity Rules](#).

- **Improvement:**

The `/supplychains/leafEntities` endpoint is now available.

This release introduces the `/supplychains/leafEntities` endpoint that you can use to define your own supply chain topology. This endpoint returns a list of 'leaf entities' related to a 'seed entity'.

For more information, see [Getting Leaf Entities from a Supply Chain Root](#).

- **Fixed Issue:**

The maximum entries returned for GET and POST methods is 500.

Starting with Turbonomic version 8.6.1, GET and POST methods return a maximum of 500 entries when run on `ServiceEntities`, `Groups`, or `BusinessAccount` endpoints. You should always check for pagination in your results.

## Turbonomic 8.6.0 API Fixes and Improvements

- **Improvement:**

The API now includes the `Plan Destinations` endpoint.

A plan destination is an external resource where plan results can be uploaded. Currently, you can upload the results of a 'Migrate to Cloud' plan to the Azure Migrate portal. Use the `Plan Destinations` endpoint to get plan destinations and run uploads.

- **Improvement:**

When deleting a reservation, you can now specify whether to permanently delete that reservation.

The API now includes the new `reservation_force_delete` parameter that you can pass along with a DELETE call to a specific reservation. When `reservation_force_delete = true`, the system removes the reservation permanently, no matter what state it is in. Before this release, a reservation that you deleted via the API or user interface continued to display for deletion, and was only permanently deleted after 48 hours.

- **Improvement:**

The API includes a new LABEL commodity type.

The `commodityType` enum includes a new LABEL commodity type.

- **Improvement:**

For API developers, when a reservation you create fails, the returned DTO includes more data about the failure.

When you create a reservation, and the reservation fails, the API returns a DTO to explain the cause of the failure. This DTO includes new fields:

- `timespan`: Indicates whether the reservation calculation was based on realtime or historical resource utilization data. Can be one of:
  - `Current`: The calculations were based on realtime utilization data
  - `Historical`: The calculations were based on historical utilization
- `cluster`: The name of the cluster that came the closest to being able to provide the resources that the reservation needed.
- `hrefCluster`: A link to stats for the closest-seller cluster.
- `hrefEntity`: A link to stats for the closest-seller entity that is a member of the closest-seller cluster.

For example, the DTO can include data similar to the following for a placement failure:

```
"placements": {
 "failureInfos": [
 {
 "resource": "StorageAmount",
 "timespan": "Historical",
 "closestSeller": {
 "uuid": "74167780552264",
 "displayName": "QSGRID01:DC6DS2",
 "className": "Storage",
 "cluster": "StorageCluster-A",
 "hrefEntity": "/reservations/stats/74167780552264",
 "hrefCluster": "/reservations/stats/74167780552265"
 },
 "maxQuantityAvailable": 248294.59024047852,
 "quantityRequested": 273184.5625,
 "units": "MB"
 }
]
}
```

- **Improvement:**

Handling of non-executable actions has been improved.

This release improves the handling of actions that are currently not executable due to prerequisite actions.

For example, in order to suspend Host A, VM\_01 in the host must first move to Host B. However, Host B only has capacity for one VM and is currently hosting VM\_02. In this case, Host A suspension is blocked by two prerequisite actions – VM\_02 moving to another host and VM\_01 moving to Host B.

The Action Details page for the main action (Host A suspension in the example) now shows the following information:

"Blocked by actions on target or destination that need to be executed first."

For earlier releases, the Action Details page indicated that the action is blocked by a policy.

For API developers, the ActionApiDTO now shows this state. Turbonomic sets the BLOCKED\_BY key of the relatedActionsCountByType parameter to 0, which indicates that an action is currently not executable due to prerequisite actions.

When all the prerequisite actions have been executed, the main action becomes executable.

- **Improvement:**

When you examine a failed reservation via the API, the data now returns a reason for the failure.

After you create a reservation, you can GET the reservations endpoint for data about the current reservations in your environment. The resulting data includes a status. If the status is PLACEMENT\_FAILED, Turbonomic could not place all the workloads in the reservation for some reason. Starting with version 8.5.4, the placements field in the data includes invalidInfo to give a reason for the placement failure. The reason can be:

- MARKET\_CONNECTIVITY\_ERROR: The reservation process could not connect to the Market component for some reason.
- MARKET\_NOT\_READY: The market does not have data from the Headroom plans for the given reservation scope.
- INVALID\_CONSTRAINTS: The reservation policies include conflicting or invalid constraints.

For example, the data for a single reservation could include the following data:

```
"demandEntities": [
 {
 "template": {
 "uuid": "215167980658768",
 "displayName": "MyVmTemplate",
 "className": "VirtualMachineProfile"
 },
 "placements": {
 "invalidInfo": {
 "invalidReason": "MARKET_CONNECTIVITY_ERROR"
 }
 }
 }
]
```

## Versions Older than 8.6.0

To download documentation for versions older than 8.6.0, see the [Archives](#).

## Reference: API Deprecation Contract

The following endpoints, requests, DTOs, and parameters are deprecated in the Turbonomic API. The changes will be finalized and removed from the API in the listed release.

**Note:**

See the instructions for alternative usage for deprecated items.

### 8.10.1

|                    |      |
|--------------------|------|
| Endpoints          | None |
| Requests           | None |
| Request Parameters | None |
| DTOs               | None |

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DTO Parameters | <ul style="list-style-type: none"> <li>• DTO: StatApiInputDTO</li> </ul> <p>Parameter Name: costPrice</p> <p>Instruction: Use the dedicated cost endpoints in /entities, /groups, and /markets instead.</p> <p><b>Note:</b></p> <p>These are marked "In Development" currently but will reach GA soon.</p> <ul style="list-style-type: none"> <li>- /entities/{entity_Uuid}/cost</li> <li>- /groups/{group_Uuid}/cost</li> <li>- /markets/{market_Uuid}/cost</li> </ul> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 8.11.1

|                    |                                                                                                                                                                                                                                   |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Endpoints          | <ul style="list-style-type: none"> <li>• GET /tags/{tagKey}/entities</li> </ul> <p>Instruction: Use GET /tags/entities. This endpoint better supports searching for certain special characters (like slash) within tag names.</p> |
| Requests           | None                                                                                                                                                                                                                              |
| Request Parameters | None                                                                                                                                                                                                                              |
| DTOs               | None                                                                                                                                                                                                                              |
| DTO Parameters     | None                                                                                                                                                                                                                              |

## Reference: API Deprecation History

The following endpoints, requests, DTOs, and/or parameters have been deprecated in the Turbonomic API. The changes were finalized and removed from the API in the listed release.

**Note:**

See the instructions for alternative usage for deprecated items.

### 8.9.1

|                    |                                                                                                                                                                             |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Endpoints          | None                                                                                                                                                                        |
| Requests           | None                                                                                                                                                                        |
| Request Parameters | None                                                                                                                                                                        |
| DTOs               | <ul style="list-style-type: none"> <li>• DTO: ScenarioChangeApiDTO</li> </ul> <p>Instruction: Use the loadChanges or configChanges parameters in the ScenarioApiDTO.</p>    |
| DTO Parameters     | <ul style="list-style-type: none"> <li>• DTO: ScenarioApiDTO</li> </ul> <p>Parameter Name: changes</p> <p>Instruction: Use the loadChanges or configChanges parameters.</p> |

### 8.8.1

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Endpoints          | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Requests           | <ul style="list-style-type: none"><li>• GET /search/market/<br/>Instruction: Use GET /markets.</li><li>• GET /search/target/<br/>Instruction: Use GET /targets.</li><li>• GET /groups/&lt;groupId&gt;/actions/&lt;actionId&gt;<br/>Instruction: Do not use this call (not implemented).</li><li>• GET, POST /target/&lt;targetId&gt;/stats<br/>Instruction: Do not use this call (not implemented).</li><li>• GET /businessunits/&lt;busUnitId&gt;/stats<br/>Instruction: Use GET /entities/&lt;entityId&gt;/stats or GET /stats/&lt;entityId&gt;/stats. These calls use the business unit entity ID.</li><li>• POST /businessunits/&lt;busUnitId&gt;/stats<br/>Instruction: Use POST /stats/&lt;entityId&gt;/stats or POST /stats.</li></ul> |
| Request Parameters | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| DTOs               | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| DTO Parameters     | None                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ENUM Values        | <ul style="list-style-type: none"><li>• entityType:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li><li>• entityTypes:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li><li>• entityTypeFilter:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li><li>• providerEntityType:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li><li>• relatedEntityType:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li><li>• providerType:VMSpec<br/>Instruction: Use VirtualMachineSpec.</li></ul>                                                                                                                                                                                                                                                 |

### 8.7.1

|           |                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Endpoints | None                                                                                                                                                                                    |
| Requests  | <ul style="list-style-type: none"><li>• DELETE /entities/&lt;entityId&gt;/tags/&lt;tagKey&gt;<br/>Instruction: Use DELETE /entities/&lt;entityId&gt;/tags?key=&lt;tagKey&gt;.</li></ul> |

|                    |                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <ul style="list-style-type: none"> <li>DELETE /groups/&lt;groupId&gt;/tags/&lt;tagKey&gt;<br/>Instruction: Use DELETE /groups/&lt;groupId&gt;/tags?key=&lt;tagKey&gt;.</li> </ul>                                                                            |
| Request Parameters | None                                                                                                                                                                                                                                                         |
| DTOs               | <ul style="list-style-type: none"> <li>LicenseApiInputDTO<br/>Instruction: Use LicenseApiDTO.</li> </ul>                                                                                                                                                     |
| DTO Parameters     | <ul style="list-style-type: none"> <li>DTO: HttpProxyDTO<br/>Parameter Name: portNumber<br/>Instruction: Use proxyPortNumber.</li> <li>DTO: LogEntryApiDTO<br/>Parameter Name: reasonCommodity<br/>Instruction: Use the array, reasonCommodities.</li> </ul> |

### 8.6.1

#### Behavior Changes:

- Pagination of Returned Data from /search requests:

For searches, the API paginates the returned data. Starting with version 8.6.1, the default pagination limit is 100 entries, and the maximum pagination limit is 500. If you do not specify a pagination limit, then the API returns pages set to the default limit of 100. If you make no settings, and your search results include more than 100 entries, the results will be paginated. Your scripts should always check for a pagination cursor when processing search results.

You can change these settings in the charts\_v1alpha1\_xl\_cr.yaml resource. Find the entries for:

- apiPaginationDefaultLimit
- apiPaginationMaxLimit

|                    |                                                                                                                                                                                                                                                                                                                                              |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Endpoints          | None                                                                                                                                                                                                                                                                                                                                         |
| Requests           | None                                                                                                                                                                                                                                                                                                                                         |
| Request Parameters | None                                                                                                                                                                                                                                                                                                                                         |
| DTOs               | <ul style="list-style-type: none"> <li>ReportScheduleApiDTO<br/>This release will remove ReportScheduleApiDTO.</li> </ul>                                                                                                                                                                                                                    |
| DTO Parameters     | <ul style="list-style-type: none"> <li>DTO: ActionApiDTO<br/>Parameter Name: CurrentLocation<br/>Instruction: The object in this parameter will only contain the following fields: <ul style="list-style-type: none"> <li>- links</li> <li>- uuid</li> <li>- className</li> <li>- displayName</li> <li>- discoveredBy</li> </ul> </li> </ul> |

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <ul style="list-style-type: none"> <li>- environmentType</li> <li>- vendorIDs</li> <li>• DTO: ActionApiDTO <ul style="list-style-type: none"> <li>Parameter Name: NewLocation</li> <li>Instruction: The object in this parameter will only contain the following fields: <ul style="list-style-type: none"> <li>- links</li> <li>- uuid</li> <li>- className</li> <li>- displayName</li> <li>- discoveredBy</li> <li>- environmentType</li> <li>- vendorIDs</li> </ul> </li> </ul> </li> <li>• DTO: ProductVersionDTO <ul style="list-style-type: none"> <li>Parameter Name: buildUser</li> <li>Instruction: Deprecated with no replacement. This parameter was not implemented in the 8.x product family.</li> </ul> </li> <li>• DTO: GroupApiDTO <ul style="list-style-type: none"> <li>Parameter Name: remoteId</li> <li>Instruction: Deprecated with no replacement. This parameter was not implemented in the 8.x product family.</li> </ul> </li> <li>• DTO: UserApiDTO <ul style="list-style-type: none"> <li>Parameter Names: <ul style="list-style-type: none"> <li>- roleUuid</li> <li>- roleName</li> </ul> </li> <li>Instruction: Deprecated. Use the roles parameter in UserApiDTO.</li> </ul> </li> </ul> |
|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### Versions Older than 8.6.0

To download documentation for versions older than 8.6.0, see the [Archives](#).



---

# Chapter 10. Integration - Data Ingestion Framework

Welcome to the Turbonomic Data Ingestion Framework (DIF). This framework enables you to define custom entities and entity metrics for your environment, and load them into the Turbonomic Market for analysis.

The DIF consists of:

- A JSON schema that declares the structure of the data you will load into the Turbonomic Market
- A probe component running on the Turbonomic platform that periodically requests your data to ingest any updates

---

## Topology Object

The Topology object is an array of entities that you want to add to the overall Turbonomic topology of managed entities. When Turbonomic loads your DIF data, it displays the entities from your topology object in the supply chain.

The Topology object includes the following properties:

|            |                                                                                                                                                                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| version    | Required String<br>The version of DIF with which your data complies. For this version, V1.                                                                                                                                                                                                           |
| updateTime | Required Integer<br>Use this to record the time that you generated the data. You should periodically generate new data to reflect changes to your environment. This value tracks when the changes were made.                                                                                         |
| scope      | Required String<br>The scope is a unique identifier for the topology you are creating. Note that the DIF probe can load multiple topologies from different data servers. The scope effectively declares a namespace that the probe can use to separate its processing of these different topologies. |
| topology   | Required Array of Entity<br>The array of entities that make up your topology.<br>See <a href="#">Entity Object</a> .                                                                                                                                                                                 |

---

## Entity Object

The Entity object describes an individual entity. You can create a stand-alone entity, or you can stitch an entity into a larger topology using relationships such as `hostedOn`, or `partOf`. An entity can include specific metrics.

An entity you create can have one of two functions in your Turbonomic supply chain:

- Topology Entity

The entity you create exists in the topology, and displays as a member of the supply chain. For example, assume you want to create a business application, and then create services that are members of that business application. You would create these as Topology Entities.

- Proxy Entity

The entity you create serves as a place-holder in the topology data you are creating. You can assign metrics to this entity, or add relationships to it to make it part of some other entity. But you also

specify the `matchIdentifiers` relationship to map this entity to an entity that already exists in the environment. The match identifier is the IP address of the entity you are mapping to.

When Turbonomic loads the topology, it moves your metric and relationship data from the topology file into the matched entity. The proxy entity does not appear in the Turbonomic supply chain, but the metric and relationship declarations do appear in the GUI for the matched entity.

The Entity object includes the following properties:

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type             | <p>Required String</p> <p>Can be one of:</p> <ul style="list-style-type: none"> <li>• <code>businessApplication</code>: a Business Application entity</li> <li>• <code>businessTransaction</code>: a Transaction entity</li> <li>• <code>service</code>: a Service entity</li> <li>• <code>databaseServer</code>: a Database Server entity</li> <li>• <code>application</code>: an Application Component entity</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| uniqueId         | <p>Required String</p> <p>A unique identifier for the entity. This identifies the entity among all other instances of that type in your topology. Because the ID must be globally unique, you should use naming conventions that express a namespace for this topology that you're creating.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| name             | <p>Required String</p> <p>The display name for the entity. This is not required to be unique, but it is usually convenient to give unique display names.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| matchIdentifiers | <p>Optional Object</p> <p>This object contains the IP address of the entity you want to match, as follows:</p> <pre>{   "ipAddress" : "11.22.33.44" }</pre> <p>If you specify <code>matchIdentifiers</code>, then this entity will be a Proxy entity. You specify the IP address of the entity that you want to modify. Then Turbonomic applies the metrics or the <code>partOf</code> specification from this proxy entity onto the matching entity.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| hostedOn         | <p>Optional Object</p> <p>The entity that is a host, or provider, for this entity. This object contains the following properties:</p> <ul style="list-style-type: none"> <li>• <code>hostEntityType</code> (Required Array of String): An array of possible entity types. The probe starts with the first entity type in the list, and checks for a matching Uuid or IP address. It uses the first one to succeed. This array can contain the strings, <code>container</code> and <code>virtualMachine</code>.</li> <li>• <code>hostUuid</code> (Optional String): The unique identifier for the host. This can come be the Uuid you have declared for some other entity in your topology, or you can use the Turbonomic Rest API to get the Uuid of a specific entity in the environment. If you provide a Uuid, you do not need to provide an IP address.</li> <li>• <code>ipAddress</code> (Optional String): The IP address of a host entity in your environment. If you provide an IP address, you do not need to provide a Uuid for the entity.</li> </ul> |
| partOf           | <p>Required Array of Objects</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|         | <p>An array of entities that contain this entity. For example, one business application can contain multiple service and transaction entities. The business application would be the <i>container</i> or <i>parent</i> entity. Also note that your entity can be contained by more than one parent.</p> <p>The objects that are members of this array identify the entity type and the Uuid of the parent, as follows:</p> <pre data-bbox="440 367 1481 493"> {   "entityType": "businessApplication",   "ipAddress" : "11.22.33.44" } </pre> <p>The entity type can be one of:</p> <ul data-bbox="440 546 1481 772" style="list-style-type: none"> <li>• businessApplication: a Business Application entity</li> <li>• businessTransaction: a Transaction entity</li> <li>• service: a Service entity</li> <li>• databaseServer: a Database Server entity</li> <li>• application: an Application Component entity</li> </ul> |
| metrics | <p>Optional Array of metricsEntry</p> <p>The list of Metric Entry objects that specify metrics and values for this entity. See <a href="#">Metrics Entry Object</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

## Commodities Bought and Sold by Entities

Entities in a Turbonomic supply chain exist in relationship to each other. One entity can sell commodities to some entities, and it can buy commodities from other entities. A commodity is the resource that is expressed by a [Metrics Entry Object](#).

There are constraints to these relationships. For example, a businessApplication cannot buy dbMem from a businessTransaction entity, but it can buy dbMem from a databaseServer entity.

The following table lists entity types, and the commodities they can buy and sell:

| Entity Type         | Commodities Sold                                                                                       | Commodities Bought                                                                                                                                                                                                                                                                                                                   |
|---------------------|--------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| businessApplication | <ul style="list-style-type: none"> <li>• kpi</li> </ul>                                                | <ul style="list-style-type: none"> <li>• Bought From: service <ul style="list-style-type: none"> <li>– transaction</li> <li>– responseTime</li> <li>– kpi</li> </ul> </li> <li>• Bought From: businessTransaction <ul style="list-style-type: none"> <li>– transaction</li> <li>– responseTime</li> <li>– kpi</li> </ul> </li> </ul> |
| businessTransaction | <ul style="list-style-type: none"> <li>• transaction</li> <li>• responseTime</li> <li>• kpi</li> </ul> | <ul style="list-style-type: none"> <li>• Bought From: service <ul style="list-style-type: none"> <li>– transaction</li> <li>– responseTime</li> <li>– kpi</li> </ul> </li> </ul>                                                                                                                                                     |
| service             | <ul style="list-style-type: none"> <li>• transaction</li> </ul>                                        | <ul style="list-style-type: none"> <li>• Bought From: application</li> </ul>                                                                                                                                                                                                                                                         |

| Entity Type    | Commodities Sold                                                                                                                                                         | Commodities Bought                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <ul style="list-style-type: none"> <li>• responseTime</li> <li>• heap</li> <li>• remainingGCCapacity</li> <li>• threads</li> <li>• kpi</li> </ul>                        | <ul style="list-style-type: none"> <li>– transaction</li> <li>– responseTime</li> <li>– heap</li> <li>– remainingGCCapacity</li> <li>– threads</li> <li>– kpi</li> <li>• Bought From: databaseServer <ul style="list-style-type: none"> <li>– transaction</li> <li>– responseTime</li> <li>– kpi</li> </ul> </li> </ul> |
| application    | <ul style="list-style-type: none"> <li>• transaction</li> <li>• responseTime</li> <li>• heap</li> <li>• remainingGCCapacity</li> <li>• threads</li> <li>• kpi</li> </ul> | <ul style="list-style-type: none"> <li>• Bought From: virtualMachine <ul style="list-style-type: none"> <li>– cpu</li> <li>– memory</li> </ul> </li> <li>• Bought From: container <ul style="list-style-type: none"> <li>– cpu</li> <li>– memory</li> </ul> </li> </ul>                                                 |
| databaseServer | <ul style="list-style-type: none"> <li>• transaction</li> <li>• responseTime</li> <li>• connection</li> <li>• dbMem</li> <li>• dbCacheHitRate</li> <li>• kpi</li> </ul>  | <ul style="list-style-type: none"> <li>• Bought From: virtualMachine <ul style="list-style-type: none"> <li>– cpu</li> <li>– memory</li> </ul> </li> </ul>                                                                                                                                                              |
| virtualMachine | <ul style="list-style-type: none"> <li>• cpu</li> <li>• memory</li> </ul>                                                                                                |                                                                                                                                                                                                                                                                                                                         |
| container      | <ul style="list-style-type: none"> <li>• cpu</li> <li>• memory</li> </ul>                                                                                                |                                                                                                                                                                                                                                                                                                                         |

## Metrics Entry Object

The Metrics Entry object expresses a single metric, with its values for utilization and capacity. You should be aware of the following points concerning metrics:

- The given metric types are all types that Turbonomic supports in its supply chain with the exception of kpi metrics.
- The kpi metric is for Key Performance Indicators. You can track any indicator you want, against a capacity that you specify. Charts will include this metric, and will show the percentage of capacity your entities are utilizing. For example, you could track the number of issues for a given application, and charts would show when the too many issues are currently open.

### Note:

These metrics do not affect Turbonomic analysis or the actions it generates.

- For native metrics that you track for your entities, charts display the values. However, these metrics do not affect Turbonomic analysis or the actions it generates.

- Not all metric types make sense for all entity types. For example, dbMem makes sense for databaseServer entities, but not for a businessTransaction entity. For more information, see [Commodities Bought and Sold by Entities](#).

Each metrics entry object is of a specific type. For each type, you express the average and capacity values in a specific metric unit. The following table describes each metric type:

|                     |                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| kpi                 | <p>Key Performance Indicator</p> <p>Metric Unit: count</p> <p>A custom metric that you can use to track entities by different criteria. For example, you can use kpi to track the number of support tickets that are open for a given application.</p> <p>The key that you provide gives the name of the metric that you will see in charts and tables in the Turbonomic user interface.</p> |
| responseTime        | <p>Response Time</p> <p>Metric Unit: ms</p> <p>A measure of the time it takes for an entity (for example, an application or a service) to process a transaction.</p>                                                                                                                                                                                                                         |
| transaction         | <p>Transactions Per Second</p> <p>Metric Unit: tps</p>                                                                                                                                                                                                                                                                                                                                       |
| connection          | <p>Concurrent Connections</p> <p>Metric Unit: count</p>                                                                                                                                                                                                                                                                                                                                      |
| heap                | <p>Amount of Heap in Use</p> <p>Metric Unit: mb</p>                                                                                                                                                                                                                                                                                                                                          |
| remainingGCCapacity | <p>Time Spent on Garbage Collection</p> <p>Metric Unit: pct</p> <p>For a 10 minute period, the percentage of that time that is spent on garbage collection.</p>                                                                                                                                                                                                                              |
| cpu                 | <p>Utilization of CPU Capacity</p> <p>Metric Unit: mhz</p> <p>The schema specifies a rawData object for this metric. The rawData is not used at this time.</p>                                                                                                                                                                                                                               |
| memory              | <p>Utilization of Memory</p> <p>Metric Unit: mb</p> <p>The schema specifies a rawData object for this metric. The rawData is not used at this time.</p>                                                                                                                                                                                                                                      |
| threads             | <p>Concurrent Threads for the Entity</p> <p>Metric Unit: count</p>                                                                                                                                                                                                                                                                                                                           |
| dbCacheHitRate      | <p>Rate of Data Requests Satisfied by Cache</p>                                                                                                                                                                                                                                                                                                                                              |

|       |                                                                                                                                                                                                                                                   |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>Metric Unit:</p> <p>Percentage of data requests that are satisfied by the database cache, where a greater value indicates fewer disk reads for data. The capacity is assumed to be 100% – You do not need to provide a value for capacity.</p> |
| dbMem | <p>Utilization of Database Memory</p> <p>Metric Unit: mb</p> <p>When a database has its own allocated capacity of memory, this tracks the utilization of that capacity.</p>                                                                       |

## Hello World

---

This tutorial walks you through a Hello World project. This is the simplest project you can deploy to show end-to-end functioning of the Data Ingestion Framework (DIF).

The DIF pairs a specific data format (a JSON schema) with a Turbonomic probe. You deploy a DIF data server and configure it as a Turbonomic target. The DIF probe gets the data your target delivers, and uses it to add entities or metric values to the Turbonomic supply chain.

This tutorial shows you how to put this all together. Successfully completing it will prove that you have a valid environment to deploy and test DIF data. In this tutorial you will:

- Deploy a server that can serve up a JSON string over HTTP or HTTPS
- Create the data to add a single entity to the Turbonomic topology
- Validate your data against the DIF JSON schema
- Configure your DIF data server as a target in Turbonomic

### Step 1: Deploy a DIF Data Server

A DIF data server is simply an HTTP or HTTPS server that can return a JSON string on request. The request must be a static URL that Turbonomic can use to GET this data at regular intervals. Just make sure the URL endpoint returns your JSON data.

For this tutorial, the JSON data is static, so you can serve it up from any web server that your Turbonomic installation can access. This tutorial assumes you already have a web server on your network that you can use. You must be able to create a new directory on that server, and upload a JSON file to it.

For this tutorial, we will assume your web server is on the domain, `MyDomain.com`, and we assume your Turbonomic instance can reach it either directly or through a proxy.

Go into the server and create the directory, `www/dif/`. Your server will serve up the JSON data from here.

Note that the DIF data server domain should be a static address. In a real-world deployment, the static address ensures the same data server is available for the long term.

#### Note:

If you don't have a web server handy, you can use the sample `node.js` that we include in the examples repository. You can deploy that on any machine in your network that has node installed. You can even install node on your Turbonomic instance, and deploy the server there.

### Step 2: Create the JSON Data

For this tutorial you will create a topology with a single Business Application entity. This entity will have the name "Hello World Tutorial."

In the `www/dif/` directory you created, create a file named `HelloWorld.json` and give it the following content.

```
{
 "version": "v1",
 "updateTime": 123,
 "scope": "Tutorial",
 "source": "",
 "topology": [
 {
 "uniqueId": "Hello_World_BusinessApp",
 "type": "businessApplication",
 "name": "Hello World Tutorial"
 }
]
}
```

A brief description of this data is in order. The data object begins with typical initial information such as version, time of latest update, and other descriptors. For a complete discussion of these fields, see the Schema Object description for [Topology Object](#).

Of these initial fields, `updateTime` is important to consider. In a real-world use case, you would periodically update your topology data to capture changes in your environment. As you do that, you should post the update time here. This could be useful for debugging in case your management process stops updating the data. For this tutorial, any arbitrary integer will do.

The topology itself is an array of entities. Each entity must include the following fields:

- `uniqueId`:

This identifies the entity among all other instances of that type in your topology. Because the ID must be globally unique, you should use naming conventions that express a namespace for this topology segment that you're creating.

- `type`:

The type of entity you are creating. For this tutorial we are creating a Business Application. DIF currently supports the following entity types:

- `businessApplication`
- `businessTransaction`
- `service`
- `databaseServer`
- `application`

- `name`:

The display name for the entity. This is not required to be unique, but it is usually convenient to give unique display names.

An entity can include other properties, such as metrics to show utilization of resources, or relationships to other entities that stitch it into the Turbonomic topology. For this tutorial we will not explore any of these properties.

After you save the file, your DIF data server should be able to serve up the JSON data. To test it, use the following command, assuming your own domain, and assuming you named the file `HelloWorld.json`:

```
curl MyDomain.com/dif/HelloWorld.json
```

The curl output should mirror the data you saved to the file.

### Step 3: Configure the DIF Data Server as a Turbonomic Target

Now that you have:

1. A web server with a dif directory
2. A DIF data file named HelloWorld.json in that directory
3. Validated the DIF data

You can set up the DIF data server as a target.

Go to your Turbonomic user interface, and navigate to **SETTINGS > Target Configuration**. From there, click **NEW TARGET** and choose the **Custom** target category. Within that category, click the **Data Ingestion Framework** target type.

Target configuration requires two fields – the URL to your DIF data (the JSON file you created) and a name for the target. Assuming you set up your web server according to the tutorial steps, you can give the following values:

- URL: `http://MyDomain.com/dif/HelloWorld.json`
- NAME: `Tutorial_Metric_Server`

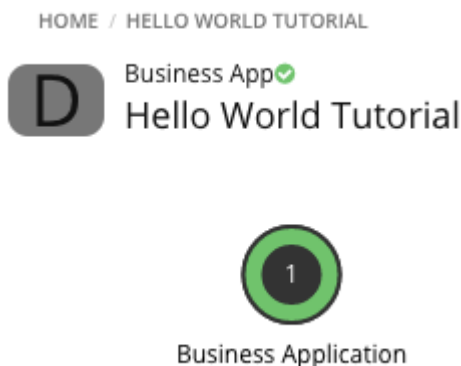
To finish the configuration, click **ADD**.

If all goes well, Turbonomic displays the Target Configuration page, and you should see the `Tutorial_Metric_Server` target in the list. It should be green, which means it successfully validated.

If the target did not validate, make sure that the URL you specified actually returns the JSON data. Also, you should ensure that the data is valid for the DIF schema.

Turbonomic requests data from targets at 10-minute intervals. Assuming your target validated, then within about 10 minutes you should see your entity in the Turbonomic supply chain.

In Turbonomic, navigate to **SEARCH** and select **Business Applications**. In the Search field, type Hello. The Hello World Tutorial entity should appear in the list. Click on that entry, and Turbonomic should change scope to show the entity in the Supply Chain.



## Wrap Up

This is the simplest possible use of the Data Ingestion Framework to load an entity into Turbonomic via a custom target. If you got this far, you know that you can produce DIF data and validate it. You know that you can deploy a DIF data server, and use it to serve up the DIF data. And you know that you can configure the data server as target.

For a real-world use case, you will want to create a fuller topology, and stitch it in with the rest of your environment. You will also want to track utilization of resources, and expose them as metrics in the DIF data.

## DIF Schema Files

---

DIF declares its data objects in two JSON schema files:

- [dif-topology-schema.json](#)



- [dif-entity-schema.json](#)

## dif-topology-schema.json

```

{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "$id": "http://turbonomic.com/dif-topology.json",
 "title": "DIF Topology File",
 "description": "Format for a DIF topology",
 "type": "object",
 "properties": {
 "version": {
 "description": "",
 "type": "string"
 },
 "updateTime": {
 "description": "Epoch timestamp for when files are generated",
 "type": "integer"
 },
 "topology": {
 "description": "List of topology entities",
 "type": "array",
 "items": {
 "$ref": "#/definitions/entity"
 },
 "minItems": 1
 }
 },
 "required": [
 "version",
 "updateTime",
 "topology"
],
 "definitions": {
 "entity": {
 "description": "Format for a DIF topology entity and metrics",
 "type": "object",
 "properties": {
 "type": {
 "description": "Entity type of the entity",
 "$ref": "dif-entity-schema.json#/definitions/entityType"
 },
 "uniqueId": {
 "description": "Unique identifier for the entity, within the scope of the target",
 "type": "string"
 },
 "name": {
 "description": "Entity name",
 "type": "string"
 },
 "matchIdentifiers": {
 "description": "Attributes used to find the entity that matches this entity",
 "type": "object",
 "properties": {
 "ipAddress": {
 "description": "IP Address of the entity used to find the matching entity",
 "type": "string"
 }
 }
 },
 "hostedOn": {
 "description": "Attributes used to find the entity that hosts this entity",
 "type": "object",
 "properties": {
 "hostType": {
 "description": "List of entity types of the possible underlying host entity. The order in which the entity types is specified is used to find the host entity.",
 "type": "array",
 "items": { "$ref": "dif-entity-schema.json#/definitions/hostEntityType" }
 },
 "hostUuid": {
 "description": "Unique identifier for the host entity",
 "type": "string"
 },
 "ipAddress": {
 "description": "IP Address of the host entity",
 "type": "string"
 }
 }
 }
 }
 }
 }
}

```

```

 },
 "required": [
 "hostType"
]
 },
 "partOf": {
 "description": "Attributes used to find all the entities that this entity is part of",
 "type": "array",
 "items": {
 "type": "object",
 "properties": {
 "entity": {
 "description": "Entity type of the parent entity ",
 "$ref": "dif-entity-schema.json#/definitions/entityType"
 },
 "uniqueId": {
 "description": "Unique identifier for the parent entity within the scope of
this target",
 "type": "string"
 }
 }
 }
 },
 "metrics": {
 "description": "List of metrics and values for the entity",
 "type": "array",
 "items": {
 "$ref": "dif-entity-schema.json#/definitions/metricsEntry"
 },
 "minItems": 1
 },
 "required": [
 "type",
 "uniqueId",
 "name"
]
}
}
}

```

## dif-entity-schema.json

```

{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "$id": "http://turbonomic.com/dif-entity.json",
 "title": "DIF Entity File",
 "description": "Format for a DIF topology entity and metrics",
 "type": "object",
 "properties": {
 "type": {
 "description": "Entity type of the entity",
 "$ref": "#/definitions/entityType"
 },
 "uniqueId": {
 "description": "Unique identifier for the entity, within the scope of the target",
 "type": "string"
 },
 "name": {
 "description": "Entity name",
 "type": "string"
 },
 "matchIdentifiers": {
 "description": "Attributes used to find the entity that matches this entity",
 "type": "object",
 "properties": {
 "ipAddress": {
 "description": "IP Address of the entity used to find the matching entity",
 "type": "string"
 }
 }
 },
 "hostedOn": {
 "description": "Attributes used to find the entity that hosts this entity",
 "type": "object",
 "properties": {
 "hostType": {

```

```

 "description": "List of entity types of the possible underlying host entity. The
order in which the entity types is specified is used to find the host entity.",
 "type": "array",
 "items": { "$ref" : "#/definitions/hostEntityType" }
 },
 "hostUuid": {
 "description": "Unique identifier for the host entity",
 "type": "string"
 },
 "ipAddress": {
 "description": "IP Address of the host entity",
 "type": "string"
 }
},
"required": [
 "hostType"
]
},
"partOf": {
 "description": "Attributes used to find all the entities that this entity is part of",
 "type": "array",
 "items": {
 "type": "object",
 "properties": {
 "entity": {
 "description": "Entity type of the parent entity ",
 "$ref": "#/definitions/entityType"
 },
 "uniqueId": {
 "description": "Unique identifier for the parent entity within the scope of this
target",
 "type": "string"
 }
 }
 }
},
},
},
"metrics": {
 "description": "List of metrics and values for the entity",
 "type": "array",
 "items": {
 "$ref": "#/definitions/metricsEntry"
 },
 "minItems": 1
}
},
"required": [
 "type",
 "uniqueId",
 "name"
],
"definitions": {
 "entityType": {
 "description": "Entity type",
 "type": "string",
 "enum": [
 "businessApplication",
 "businessTransaction",
 "service",
 "databaseServer",
 "application"
]
 },
 "hostEntityType": {
 "description": "Entity type of the host that underlies this entity",
 "type": "string",
 "enum": [
 "container",
 "virtualMachine"
]
 },
 "metricUnit": {
 "description": "Units for different metrics",
 "type": "string",
 "enum": ["count", "tps", "ms", "mb", "mhz", "pct"]
 },
 "metricValue": {
 "type": "object",
 "properties": {
 "average": { "type": "number" },
 "min": { "type": "number" },

```

```

 "max": { "type": "number" },
 "capacity": { "type": "number" },
 "key": { "type": "string" },
 "resizable": { "type": "boolean" },
 "unit": { "$ref": "#/definitions/metricUnit" }
 },
 "required": [
 "average", "unit"
]
},
"metricsEntry": {
 "description": "Supported metrics, TODO - add schemas for other supported metrics",
 "type": "object",
 "oneOf": [
 { "$ref": "#/definitions/_responseTime" },
 { "$ref": "#/definitions/_transaction" },
 { "$ref": "#/definitions/_connection" },
 { "$ref": "#/definitions/_heap" },
 { "$ref": "#/definitions/_remainingGCCapacity" },
 { "$ref": "#/definitions/_kpi" },
 { "$ref": "#/definitions/_cpu" },
 { "$ref": "#/definitions/_memory" },
 { "$ref": "#/definitions/_threads" },
 { "$ref": "#/definitions/_dbCacheHitRate" },
 { "$ref": "#/definitions/_dbMem" }
]
},
"_responseTime": {
 "description": "Specification for Response Time metrics",
 "type": "object",
 "properties": {
 "responseTime": {
 "$ref": "#/definitions/metricValue"
 }
 },
 "required": [
 "responseTime"
]
},
"_transaction": {
 "description": "Specification for Transaction metrics",
 "type": "object",
 "properties": {
 "transaction": {
 "$ref": "#/definitions/metricValue"
 }
 },
 "required": [
 "transaction"
]
},
"_connection": {
 "description": "Specification for Connections metrics",
 "type": "object",
 "properties": {
 "connection": {
 "$ref": "#/definitions/metricValue"
 }
 },
 "required": [
 "connection"
]
},
"_heap": {
 "description": "Specification for Heap metrics",
 "type": "object",
 "properties": {
 "heap": {
 "$ref": "#/definitions/metricValue"
 }
 },
 "required": [
 "heap"
]
},
"_remainingGCCapacity": {
 "description": "Specification for Collection Time metrics",
 "type": "object",
 "properties": {
 "remainingGCCapacity": {
 "$ref": "#/definitions/metricValue"
 }
 }
}

```

```

 }
 },
 "required": [
 "remainingGCCapacity"
]
},
"-threads" : {
 "description": "Specification for Threads metrics",
 "type": "object",
 "properties": {
 "threads": {
 "type": "array",
 "items": {
 "$ref": "#/definitions/metricValue"
 },
 "uniqueItems": true,
 "minItems": 1
 }
 }
},
"required": [
 "threads"
]
},
"-dbCacheHitRate" : {
 "description": "Specification for DB Cache Hit Rate metrics",
 "type": "object",
 "properties": {
 "dbCacheHitRate": {
 "type": "array",
 "items": {
 "$ref": "#/definitions/metricValue"
 },
 "uniqueItems": true,
 "minItems": 1
 }
 }
},
"required": [
 "dbCacheHitRate"
]
},
"-dbMem" : {
 "description": "Specification for DB Memory metrics",
 "type": "object",
 "properties": {
 "dbMem": {
 "type": "array",
 "items": {
 "$ref": "#/definitions/metricValue"
 },
 "uniqueItems": true,
 "minItems": 1
 }
 }
},
"required": [
 "dbMem"
]
},
"-cpu" : {
 "description": "Specification for CPU metrics",
 "type": "object",
 "properties": {
 "cpu": {
 "$ref": "#/definitions/metricValueWithRawData"
 }
 }
},
"required": ["cpu"]
},
"-memory" : {
 "description": "Specification for Memory metrics",
 "type": "object",
 "properties": {
 "memory": {
 "$ref": "#/definitions/metricValueWithRawData"
 }
 }
},
"required": ["memory"]
},
"-kpi" : {
 "description": "Specification for the custom metrics",
 "type": "object",
 "properties": {
 "kpi": {

```



---

# Chapter 11. Integration - Data Exporter

The Turbonomic platform includes the Data Exporter, which you can choose to enable when you install the platform. The Data Exporter streams Turbonomic data about entities and actions as JSON documents that it then publishes to a Kafka topic.

The exporter publishes at different frequencies, depending on the type of data:

- Entity data: Published every 10 minutes.
- Actions data:
  - Pending actions: Published every 6 hours. You can configure this via the `actionExtractionIntervalMins` variable.
  - Executed actions: Published immediately after the action completes, either with success or failure.

You can load the exported data into search and analytics services such as Elasticsearch, and use those services to visualize or analyze the entities and actions in your Turbonomic environment.

With the Data Exporter you can build custom tracking of changes to your environment, and follow the details as Turbonomic keeps your applications healthy.

---

## Enabling the Data Exporter

To support the Data Exporter, Turbonomic provides an extractor component that can stream data to a standard format. You can load that data into search and analytics services such as Elasticsearch.

To enable the Data Exporter, you must:

- Enable the extractor component.

The extractor is a component that runs as part your Turbonomic installation. The extractor is not enabled by default.

- Deploy a connector that delivers the extractor's stream to your data service.

The extractor publishes Turbonomic data as Kafka topics. The connector enables your data service to consume the data topic. This document includes a deployment file for a sample Elasticsearch connector.

## Enabling the Extractor Component

The first step to enabling the Data Exporter is to enable the extractor component. To enable the extractor:

1. Open an SSH terminal session to your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

Username: turbo

Password: [your\_private\_password]

2. Edit the `cr.yaml` file to enable the extractor component.

In the same SSH session, open the `cr.yaml` file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/
charts_v1alpha1_xl_cr.yaml
```

3. Edit the entry for the extractor component.

**Note:**

If you have enabled Embedded Reporting, then the extractor component will already be enabled (set to `true`).

You should understand that it is possible to enable the Data Exporter without enabling Embedded Reports, just as it is possible to enable Embedded Reports without enabling the Data Exporter. Search for the extractor entry in the `cr.yaml` file. It should appear as:

```
extractor:
 enabled: false
```

Change the entry to `true`.

4. Edit the entry for the extractor properties.

Search for the extractor entry in the `cr.yaml` file. It should appear as:

```
properties:
 extractor:
 enableDataExtraction: false
```

Change the entry to `true`.

5. Save and apply your changes to the platform.

After you save your changes, use `kubectl` to apply the changes:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

6. Verify that the extractor component is running.

Give the platform enough time to restart the components. Then execute the command:

```
kubectl get pods -n turbonomic
```

You should see output similar to the following:

| NAME                       | READY | STATUS  | RESTARTS |
|----------------------------|-------|---------|----------|
| ...                        |       |         |          |
| extractor-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                        |       |         |          |

Look for an entry for the `extractor` component. If the entry is present, ready, and running, then the extractor component is installed and running.

## Deploying a Connector

The extractor publishes Turbonomic data as Kafka topics. To load this data into a search and analysis service, you must deploy a connector to that service. For example, if you want to load the data into Elasticsearch, then you must deploy an Elasticsearch connector.

You deploy the connector in the same Kubernetes node that runs the Turbonomic platform. To do this, create a Kubernetes *Deployment* that declares the pods you need for the connector. Below, you can see a sample deployment of a connector to Elasticsearch.

To deploy the connector, you create a deployment `yaml` file on the same host that is running the extractor component, and execute the command:

```
kubectl create -f <MyConnectorDeployment.yaml>
```

Where `<MyConnectorDeployment.yaml>` is the name of the deployment file.

Assume the name of the deployed pod is `es-kafka-connect`. To verify that the connector is running, execute `kubectl get pods -n turbonomic`. You should see output similar to:

| NAME                              | READY | STATUS  | RESTARTS |
|-----------------------------------|-------|---------|----------|
| ...                               |       |         |          |
| es-kafka-connect-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                               |       |         |          |



After you deploy the connector, wait for a cycle of Turbonomic analysis (approximately ten minutes). Then you should be able to see the entities and actions from your Turbonomic environment, loaded as JSON in your data service.

## Connector Deployment Example

Assume that you want to deploy a connector to Elasticsearch so that service can process the exported data. For example, you could use Kibana with Elasticsearch to display data dashboards. Let's say you have:

- Deployed Elasticsearch to a VM on the network where you are running Turbonomic. The Elasticsearch host is visible from the Turbonomic Kubernetes node. You will specify this host address in the connector deployment.
- Set up an Elasticsearch index to load the Turbonomic data. You will specify this index in the connector deployment.

The following listing is a deployment that uses a Logstash image to collect the extractor data and pipe it to the Elasticsearch host. The deployment also sets up storage volumes, configures the input from the extractor, and configures output to the Elasticsearch instance.

As you go over the listing, pay attention to the following:

- The location of the Elasticsearch host and the login credentials:

```
...
 env:
 - name: ES_HOSTS
 value: "<urltoelasticsearchhost>"
 - name: ES_USER
 value: "<myelasticsearchuser>"
 - name: ES_PASSWORD
 valueFrom:
 secretKeyRef:
 name: <MyES_KeyName>
 key: <MyES_Key>
...

```

Logstash will use the following environment variables:

- ES\_HOSTS: to identify where to pipe the exported data.
  - ES\_USER: to identify the user account on Elasticsearch.
  - ES\_PASSWORD: for the account login. This connector example assumes that you have stored the Elasticsearch password as a Kubernetes Secret.
- The name of the Kafka topic:

```
...
 output {
 elasticsearch {
 index => "<MyElasticsearchIndex>"
 hosts => ["${ES_HOSTS}"]
 }
 }
...

```

The Logstash input configuration expects a single topic named `turbonomic.exporter`.

- The Logstash output configuration is to the Elasticsearch server that is identified by the ES\_HOSTS environment variable. You specify your own Elasticsearch index in place of `<MyElasticsearchIndex>`.

```
... output { elasticsearch { index => "<MyElasticsearchIndex>" hosts => ["${ES_HOSTS}"] } } ...
```

## Sample Listing: Elasticsearch Connector

This listing is a sample of a deployment file that can work to create an Elasticsearch connector for the Data Exporter. Note that you will need to change some settings, such as username and password. You also might need to specify ports and other settings to make the connector comply with your specific environment.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: elasticsearch-kafka-connect
 labels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
spec:
 replicas: 1
 selector:
 matchLabels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
 template:
 metadata:
 labels:
 app.kubernetes.io/name: elasticsearch-kafka-connect
 spec:
 containers:
 - name: logstash
 image: docker.elastic.co/logstash/logstash:7.10.1
 ports:
 - containerPort: 25826
 env:
 - name: ES_HOSTS
 value: "<UrlToMyElasticsearchHost>"
 - name: ES_USER
 value: "<MyElasticsearchUser>"
 - name: ES_PASSWORD
 valueFrom:
 secretKeyRef:
 name: <MyES_KeyName>
 key: <MyES_Key>
 resources:
 limits:
 memory: 4Gi
 volumeMounts:
 - name: config-volume
 mountPath: /usr/share/logstash/config
 - name: logstash-pipeline-volume
 mountPath: /usr/share/logstash/pipeline
 volumes:
 - name: config-volume
 configMap:
 name: logstash-configmap
 items:
 - key: logstash.yml
 path: logstash.yml
 - name: logstash-pipeline-volume
 configMap:
 name: logstash-configmap
 items:
 - key: logstash.conf
 path: logstash.conf

 apiVersion: v1
 kind: ConfigMap
 metadata:
 name: logstash-configmap
 data:
 logstash.yml: |
 http.host: "0.0.0.0"
 path.config: /usr/share/logstash/pipeline
 logstash.conf: |
 input {
 kafka {
 topics => ["turbonomic.exporter"]
 bootstrap_servers => "kafka:9092"
 client_id => "logstash"
 group_id => "logstash"
 codec => "json"
 type => "json"
 session_timeout_ms => "60000" # Rebalancing if consumer is found dead
 request_timeout_ms => "70000" # Resend request after 70 seconds
```

```

 }
 }
 filter {
 }
 output {
 elasticsearch {
 index => "<MyElasticsearchIndex>"
 hosts => ["${ES_HOSTS}"]
 user => "${ES_USER}"
 password => "${ES_PASSWORD}"
 }
 }
}

apiVersion: v1
kind: Service
metadata:
 labels:
 app: elasticsearch-kafka-connect
 name: elasticsearch-kafka-connect
spec:
 ports:
 - name: "25826"
 port: 25826
 targetPort: 25826
 selector:
 app: elasticsearch-kafka-connect

```

## Increasing the Elasticsearch Field Limit

For exports to Elasticsearch, it is possible that you will encounter the following error:

Elasticsearch: Limit of total fields [1000] has been exceeded

In the connector's logstash logs, you might see an error similar to the following:

```

[2021-05-28T17:16:15,252][INFO][o.e.a.b.TransportShardBulkAction] [node1] [turbonomic6][0]
mapping update rejected by primary
java.lang.IllegalArgumentException: Limit of total fields [1000] has been exceeded
 at org.elasticsearch.index.mapper.MappingLookup.checkFieldLimit(MappingLookup.java:200)
~[elasticsearch-7.12.0.jar:7.12.0]
 at org.elasticsearch.index.mapper.MappingLookup.checkLimits(MappingLookup.java:192)
~[elasticsearch-7.12.0.jar:7.12.0]
 at org.elasticsearch.index.mapper.DocumentMapper.validate(DocumentMapper.java:152)
~[elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.index.mapper.MapperService.newDocumentMapper(MapperService.java:429)
~[elasticsearch-7.12.0.jar:7.12.0]
 at org.elasticsearch.index.mapper.MapperService.applyMappings(MapperService.java:407)
~[elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.index.mapper.MapperService.mergeAndApplyMappings(MapperService.java:390)
~[elasticsearch-7.12.0.jar:7.12.0]
 at org.elasticsearch.index.mapper.MapperService.merge(MapperService.java:384)
~[elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.bulk.TransportShardBulkAction.executeBulkItemRequest(TransportShardBulk
Action.java:281) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.bulk.TransportShardBulkAction$2.doRun(TransportShardBulkAction.java:164
) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.common.util.concurrent.AbstractRunnable.run(AbstractRunnable.java:26)
[elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.bulk.TransportShardBulkAction.performOnPrimary(TransportShardBulkAction
.java:209) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.bulk.TransportShardBulkAction.dispatchedShardOperationOnPrimary(Transpo
rtShardBulkAction.java:115) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.bulk.TransportShardBulkAction.dispatchedShardOperationOnPrimary(Transpo
rtShardBulkAction.java:74) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.action.support.replication.TransportWriteAction$1.doRun(TransportWriteAction.j
ava:168) [elasticsearch-7.12.0.jar:7.12.0]
 at
org.elasticsearch.common.util.concurrent.ThreadContext$ContextPreservingAbstractRunnable.doRun(T
hreadContext.java:732) [elasticsearch-7.12.0.jar:7.12.0]

```

```

at
org.elasticsearch.common.util.concurrent.AbstractRunnable.run(AbstractRunnable.java:26)
[elasticsearch-7.12.0.jar:7.12.0]
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128) [?:?]
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628) [?:?]
at java.lang.Thread.run(Thread.java:834) [?:?]

```

If you encounter this error, Elastic search might not ingest the full set of data. For example, Elasticsearch might not show some entities that you can see in the Turbonomic user interface.

This error occurs when the number of fields in a single JSON object exceeds the default limit of 1000. For information about this limit, see [Mapping limit settings](#) in the Elasticsearch documentation.

To correct this issue, you can increase the field limit:

- In Kibana:

Go to **Index Management > Settings** and increase the limit. For more information, see [Index management](#) in the Elasticsearch documentation.

- Directly in Elasticsearch

You can execute the following command:

```

curl -X PUT -H "Content-Type: application/json" -d "{\"index.mapping.total_fields.limit\":
2000}" \
http://{elasticsearch_ip}:9200/{index_name}/_settings

```

Where `{elasticsearch_ip}` is the IP address of your Elasticsearch, and `{index_name}` is the name of the index that exceeds the field limit.

The extractor publishes Turbonomic data as Kafka topics. To load this data into a search and analysis service, you must deploy a connector to that service. For example, if you want to load the data into Elasticsearch, then you must deploy an Elasticsearch connector.

You deploy the connector in the same Kubernetes node that runs the Turbonomic platform. To do this, create a Kubernetes *Deployment* that declares the pods you need for the connector. For an example of an `ElasticSearch` connector, see [Connector Deployment Example](#)

To deploy the connector, you create a deployment yaml file on the same host that is running the extractor component, and execute the command:

```
kubectl create -f <MyConnectorDeployment.yaml>;
```

Where `<MyConnectorDeployment.yaml>` is the name of the deployment file.

Assume the name of the deployed pod is `es-kafka-connect`. To verify that the connector is running, execute `kubectl get pods -n turbonomic`. You should see output similar to:

| NAME                              | READY | STATUS  | RESTARTS |
|-----------------------------------|-------|---------|----------|
| ...                               |       |         |          |
| es-kafka-connect-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                               |       |         |          |

After you deploy the connector, wait for a cycle of Turbonomic analysis (approximately ten minutes). Then you should be able to see the entities and actions from your Turbonomic environment, loaded as JSON in your data service.

## Configuring Data Export Intervals

By default, the Data Exporter sends JSON data for entities and groups at 10-minute intervals. This usually gives a good flow of data that your data search and analysis services can ingest.

For very large environments, it is possible that the quantity of data can overwhelm the ability of your service to ingest it. In the worst case, this can result in Out Of Memory errors in your service connector. If this occurs, lengthening the export interval can reduce the overall volume of data your service has to ingest.

You can configure the data export interval in the extractor component for action data and for entity and group data.

1. Open an SSH terminal session to your Turbonomic instance.  
Log in with the System Administrator that you set up when you installed Turbonomic:

Username: turbo

Password: [your\_private\_password]

2. Edit the cr.yaml file to add export intervals to the extractor component.

In the same SSH session, open the cr.yaml file for editing. For example:

```
vi /opt/turbonomic/kubernetes/operator/deploy/crds/
charts_v1alpha1_x1_cr.yaml
```

3. Edit the entry for the extractor properties.

Search for the extractor properties entry in the cr.yaml file. It should appear as:

```
properties:
 extractor:
 enableDataExtraction: true
```

4. Add the properties to control the export intervals.

By default, the extractor exports entity and group data every 10 minutes, and action data every 6 hours. You can add properties to the extractor to change the interval. Each setting is in units of minutes.

With these settings, you can specify intervals:

- Globally

`globalExtractionIntervalMins`

If you want to set a single interval for all extractor data exports (entities, groups, and actions), use this property.

- For Entities and Groups

`entityExtractionIntervalMins`

To specify an export interval for entity and group data, use this property. If it is present in the extractor properties, this setting takes precedence for entities and groups.

- For Actions

`actionExtractionIntervalMins`

To specify an export interval for actions data, use this property. If it is present in the extractor properties, this setting takes precedence for actions.

For example, the following specification sets a 30-minute interval on entities and groups, and a one-hour interval on actions:

```
properties:
 extractor:
 enableDataExtraction: true
 entityExtractionIntervalMins: 30
 actionExtractionIntervalMins: 60
```

5. Save and apply your changes to the platform.

After you save your changes, use `kubectl` to apply the changes:

```
kubectl apply -f \
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_x1_cr.yaml
```

6. Restart the extractor component.

First, get the name of the extractor pod. Execute the command:

```
kubectl get pods -n turbonomic
```

You should see output similar to the following:

| NAME                       | READY | STATUS  | RESTARTS |
|----------------------------|-------|---------|----------|
| ...                        |       |         |          |
| extractor-5f41dd61c4-4d61q | 1/1   | Running | 0        |
| ...                        |       |         |          |

Search the listing for the extractor pod, and use that as the pod name. Then delete the pod to force it to restart. For the example above, enter the command:

```
kubectl delete pod extractor-5f41dd61c4-4d61q
```

7. Verify that the extractor component is running.

Give the platform enough time to restart the components. Then execute the command:

```
kubectl get pods -n turbonomic
```

You should see the extractor pod in the listing, and its status should be RUNNING.

## Entity Object

Every 10 minutes, the extractor sends JSON data to Kafka for individual entities. Each entity object has two top-level fields:

- `timestamp`: When the entity was broadcast
- `entity`: The object that describes the given entity.

### Sample Entity Object

```
{
 "timestamp": "2021-01-15T19:17:13",
 "entity": {
 "environment": "ON_PREM",
 "name": "test-rdm-shared2",
 "attrs": {
 "guest_os_type": "WINDOWS",
 "connected_networks": [
 "VM Network"
],
 "num_cpus": 1
 },
 "state": "POWERED_OFF",
 "oid": "73754533904736",
 "type": "VIRTUAL_MACHINE",
 "related": {
 "PHYSICAL_MACHINE": [
 {
 "name": "hp-esx46.corp.vmturbo.com",
 "oid": "73754533901584"
 }
],
 "COMPUTE_CLUSTER": [
 {
 "name": "Datacenter DC7/Cluster01",
 "oid": "284907641559697"
 }
],
 "DATACENTER": [
 {
 "name": "Datacenter DC7",
 "oid": "73754533900625"
 }
],
 "VIRTUAL_DATACENTER": [
 {
 "name": "Resources-Cluster01",
 "oid": "73754533901777"
 }
],
 "VIRTUAL_VOLUME": [
```

```

 {
 "name": "Vol-test-rdm-shared2-QSGRID01:DC7DS2",
 "oid": 73754533909440
 },
 {
 "name": "Vol-test-rdm-shared2-test-rdm-shared1 : [QSGRID01:DC7DS2] test-rdm-shared1/
test-rdm-shared1_1.vmdk",
 "oid": 73754533909439
 }
],
 "STORAGE": [
 {
 "name": "test-rdm-shared1 : [QSGRID01:DC7DS2] test-rdm-shared1/test-rdm-
shared1_1.vmdk",
 "oid": 73754533901780
 },
 {
 "name": "QSGRID01:DC7DS2",
 "oid": 73754533900756
 }
],
 "DISK_ARRAY": [
 {
 "name": "test-rdm-shared1",
 "oid": 73754533901786
 },
 {
 "name": "DiskArray-QSGRID01:DC7DS2",
 "oid": 73754533900667
 }
]
},
"metric": {
 "STORAGE_AMOUNT": {
 "consumed": 3
 },
 "MEM_ALLOCATION": {
 "consumed": 0
 },
 "CPU_PROVISIONED": {
 "consumed": 2600
 },
 "VCPU": {
 "current": 0,
 "capacity": 2600,
 "utilization": 0
 },
 "BALLOONING": {
 "consumed": 0
 },
 "STORAGE_PROVISIONED": {
 "consumed": 15367.5751953125
 },
 "SWAPPING": {
 "consumed": 0
 },
 "IO_THROUGHPUT": {
 "consumed": 0
 },
 "MEM": {
 "consumed": 0
 },
 "NET_THROUGHPUT": {
 "consumed": 0
 },
 "MEM_PROVISIONED": {
 "consumed": 4194304
 },
 "VMEM": {
 "current": 0,
 "capacity": 4194304,
 "utilization": 0
 },
 "STORAGE_ACCESS": {
 "consumed": 0
 },
 "CPU_ALLOCATION": {
 "consumed": 0
 },
 "STORAGE_LATENCY": {
 "consumed": 0
 }
},

```

```

 "CPU": {
 "consumed": 0
 }
 }
}

```

## Common Entity Object Fields

Every entity object includes the following fields:

| JSON Object     | Description                                                                                              | Type                             | Repeated                    | MapKeyType                  | Reference                        |
|-----------------|----------------------------------------------------------------------------------------------------------|----------------------------------|-----------------------------|-----------------------------|----------------------------------|
| accountExpenses |                                                                                                          | <a href="#">AccountExpenses</a>  | False                       |                             | <a href="#">AccountExpenses</a>  |
| attrs           | Additional descriptions for a given entity. The values of this are different for different entity types. | <a href="#">entity.attrs</a>     | False                       |                             | <a href="#">entity.attrs</a>     |
| cost            |                                                                                                          | <a href="#">EntityCost</a>       | False                       |                             | <a href="#">EntityCost</a>       |
| environment     | The environment type - Hybrid, Cloud, On-prem, etc.                                                      | <a href="#">environment_type</a> | False                       |                             | <a href="#">environment_type</a> |
| metric          | For the given metric type, the utilization and capacity of the commodity.                                | <a href="#">Commodity</a>        | False                       | <a href="#">metric_type</a> | <a href="#">Commodity</a>        |
| name            | The entity name.                                                                                         | String                           | False                       |                             |                                  |
| oid             | The entity ID.                                                                                           | Long                             | False                       |                             |                                  |
| related         | For the current entity, another entity that is connected to it in the supply chain.                      | <a href="#">RelatedEntity</a>    | <a href="#">entity_type</a> | True                        | <a href="#">RelatedEntity</a>    |
| state           | The current state of the entity - POWERED_ON, SUSPENDED, MAINTENANCE, etc.                               | <a href="#">entity_state</a>     | False                       |                             | <a href="#">entity_state</a>     |
| type            | The entity type.                                                                                         | <a href="#">entity_type</a>      | False                       |                             | <a href="#">entity_type</a>      |

### Entity Types:

Turbonomic can export data for the following types of entities.



## Metric Types:

Turbonomic tracks different metrics for different types of entities. For a given entity object, you will get the metric data that is appropriate for that entity. This list is all the metric types that the Data Exporter supports.

## Related Entity Types:

Related entities are entities that are connected to the given entity in the supply chain. For example, a VM entity can have containers and applications running on it, while it uses disk space from related storage. For a given object, the data includes all the related entities. The supported related types can be any of the supported entities, or any of the group types listed here.

## Entity Cost Data

---

The JSON for an entity can include cost data. This section describes the cost data as it relates to an entity.

### Sample Cost Data

This listing shows a sample entity object, with cost data included. Note that to make it easier to focus on cost, this listing

```
{
 "timestamp" : "2021-05-21T19:59:33",
 "entity": {
 "oid": 73978235908269,
 "name": "jjbosvm4-aws",
 "state": "POWERED_ON",
 "environment": "CLOUD",
 "type": "VIRTUAL_MACHINE",
 "cost": {
 "unit": "$/h",
 "total": 0.0070848083,
 "category": {
 "ON_DEMAND_COMPUTE": {
 "total": 0.0,
 "source": {
 "ON_DEMAND_RATE": 0.0126,
 "RI_INVENTORY_DISCOUNT": -0.0126
 }
 },
 "RESERVED_LICENSE": {
 "total": 0.0,
 "source": {
 "RI_INVENTORY_DISCOUNT": 0.0
 }
 },
 "STORAGE": {
 "total": 0.00078480854,
 "source": {
 "ON_DEMAND_RATE": 0.00078480854
 }
 },
 "RI_COMPUTE": {
 "total": 0.0063,
 "source": {
 "UNCLASSIFIED": 0.0063
 }
 }
 }
 }
 },
 "metric": {
 ...
 },
 "related": {
 ...
 },
 "attrs": {
 ...
 }
}
```

```
}
}
```

## entity.cost Fields

The cost field contains the following data:

- **unit**: The unit of the cost value; for example, \$/h.
- **total**: The total cost for the given entity.
- **category**: A cost breakdown by category. One entity can include one or more of these categories:
  - ON\_DEMAND\_COMPUTE
  - ON\_DEMAND\_LICENSE
  - RESERVED\_LICENSE
  - STORAGE
  - RI\_COMPUTE
  - IP
  - SPOT

Each category can include the following fields to give a total for that category, and to itemize the costs that contribute to that total:

- **total**: The total cost for given category.
- **source**: A breakdown of the total by source. The source can include the following types:
  - ON\_DEMAND\_RATE
  - RI\_INVENTORY\_DISCOUNT
  - BUY\_RI\_DISCOUNT
  - UNCLASSIFIED

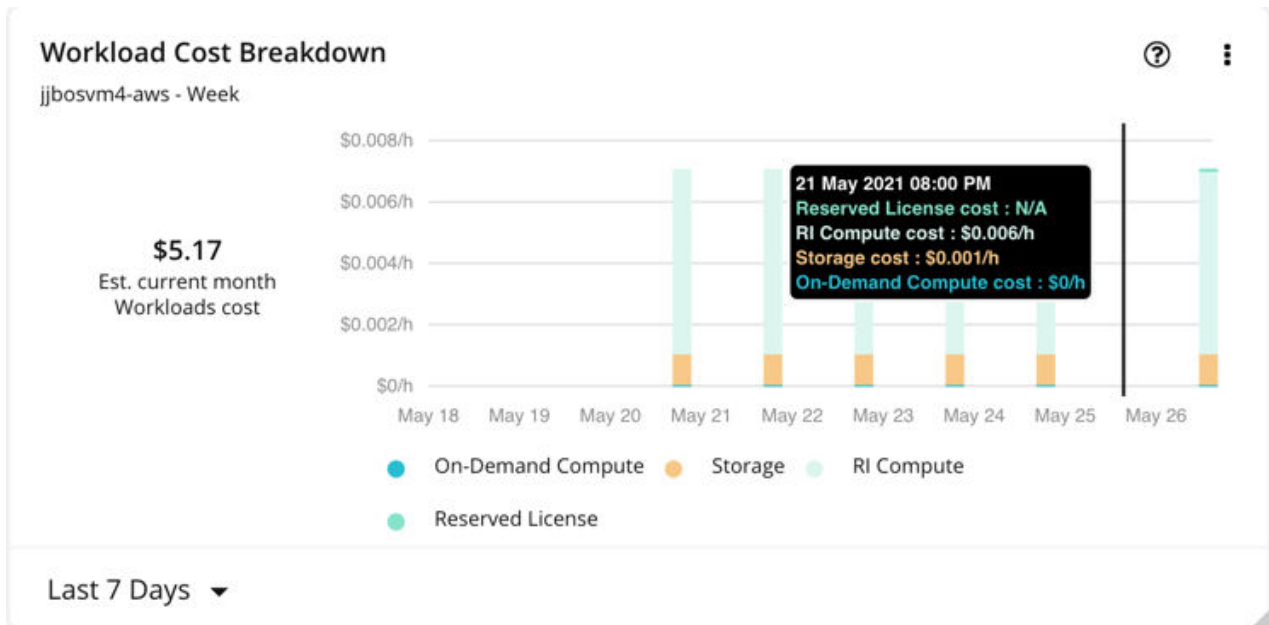
Each source type gives the cost associated with that type. For example, the sample listing includes the following for on-demand compute costs:

```
"ON_DEMAND_COMPUTE": {
 "total": 0.0,
 "source": {
 "ON_DEMAND_RATE": 0.0126,
 "RI_INVENTORY_DISCOUNT": -0.0126
 }
}
```

This gives two sources for the category cost. The total is zero because the ON\_DEMAND\_RATE is canceled out by the RI\_INVENTORY\_DISCOUNT.

The cost in the sample listing matches this Workload Cost Breakdown chart:

- The total of \$5.17 is the product of `entity.cost.total` (which is \$0.0070848083 per hour) times the hours per month (730).
- The cost breakdown in the chart matches the the costs for the same categories; ON\_DEMAND\_COMPUTE, STORAGE, RI\_COMPUTE, and RESERVED\_LICENSE.



The cost total and the breakdown by category give you the data you need to replicate the charts in Turbonomic. The breakdown by source also exposes how the costs are calculated, which is extra information that does not display in the user interface.

## Action Object

The extractor sends action data to Kafka:

- Every 6 hours for pending actions.
- Immediately after an action completes, whether completion is success or failure.

Each action object has two top-level fields:

- `timestamp`: When the action object was broadcast.
- `action`: The data that describes the given action.

### Sample Action Object

```
{
 "timestamp": "2020-12-07T18:20:10Z",
 "action": {
 "oid": "144118995513593",
 "creationTime": "2020-12-07T18:13:10Z",
 "type": "MOVE",
 "state": "READY",
 "mode": "MANUAL",
 "category": "PERFORMANCE_ASSURANCE",
 "severity": "CRITICAL",
 "description": "Move Virtual Machine hp-esx59 from hp-bl433.eng.vmturbo.com to hp-bl435.eng.vmturbo.com",
 "explanation": "Q2 VCPU Congestion",
 "target": {
 "oid": 73741823907376,
 "name": "hp-esx59",
 "type": "VirtualMachine"
 },
 "moveInfo": {
 "from": {
 "oid": "73741823904848",
 "name": "hp-bl433.eng.vmturbo.com",
 "type": "PHYSICAL_MACHINE"
 },
 "to": {
 "oid": "73741823904864",
 "name": "hp-bl435.eng.vmturbo.com",
 "type": "PHYSICAL_MACHINE"
 }
 }
 }
}
```



| JSON Object  | Description                                                                   | Type                            | Repeated | Format                   | MapKeyType                  | Reference                       |
|--------------|-------------------------------------------------------------------------------|---------------------------------|----------|--------------------------|-----------------------------|---------------------------------|
| creationTime | When analysis created the action.                                             | String                          | False    | yyyy-MM-dd'T'HH:mm:ss'Z' |                             |                                 |
| deleteInfo   | A description of a file is unattached and should be deleted.                  | <a href="#">DeleteInfo</a>      | False    |                          |                             | <a href="#">DeleteInfo</a>      |
| description  | The description that analysis generates for the action.                       | String                          | False    |                          |                             |                                 |
| explanation  | The condition that the action is addressing. For example, Q2 VCPU CONGESTION. | String                          | False    |                          |                             |                                 |
| mode         | The mode for the action execution.                                            | <a href="#">action_mode</a>     | False    |                          |                             | <a href="#">action_mode</a>     |
| moveInfo     | Information to describe a MOVE action.                                        | <a href="#">MoveChange</a>      | False    |                          |                             | <a href="#">MoveChange</a>      |
| oid          | The action object identifier.                                                 | Long                            | False    |                          |                             |                                 |
| related      | An array of entities or groups that are related to the target entity.         | <a href="#">RelatedEntity</a>   | True     |                          | <a href="#">entity_type</a> | <a href="#">RelatedEntity</a>   |
| resizeInfo   | Information to describe a RESIZE action.                                      | <a href="#">CommodityChange</a> | False    |                          |                             | <a href="#">CommodityChange</a> |
| savings      | The cost change for the action.                                               | <a href="#">CostAmount</a>      | False    |                          |                             | <a href="#">CostAmount</a>      |

| JSON Object | Description                                                        | Type                                 | Repeated | Format | MapKeyType | Reference                            |
|-------------|--------------------------------------------------------------------|--------------------------------------|----------|--------|------------|--------------------------------------|
| scaleInfo   | Information to describe a SCALE action.                            | <a href="#">MoveChange</a>           | False    |        |            | <a href="#">MoveChange</a>           |
| severity    | For a given action, the severity of the issue that it addresses.   | <a href="#">severity</a>             | False    |        |            | <a href="#">severity</a>             |
| state       | The different states for the action, such as completed or pending. | <a href="#">action_state</a>         | False    |        |            | <a href="#">action_state</a>         |
| target      | The entity that the action operates on.                            | <a href="#">ActionImpactedEntity</a> | False    |        |            | <a href="#">ActionImpactedEntity</a> |
| type        | The action type.                                                   | <a href="#">action_type</a>          | False    |        |            | <a href="#">action_type</a>          |

## Enums

---

The exported data can include the following enums.

### AttachmentState

- ATTACHED:
- UNATTACHED:

### OSType

- LINUX:
- LINUX\_WITH\_SQL\_ENTERPRISE:
- LINUX\_WITH\_SQL\_STANDARD:
- LINUX\_WITH\_SQL\_WEB:
- RHEL:
- SUSE:
- UBUNTU\_PRO:
- UNKNOWN\_OS:
- WINDOWS:
- WINDOWS\_BYOL:
- WINDOWS\_WITH\_SQL\_ENTERPRISE:
- WINDOWS\_WITH\_SQL\_STANDARD:
- WINDOWS\_WITH\_SQL\_WEB:

## **action\_category**

- COMPLIANCE:
- EFFICIENCY\_IMPROVEMENT:
- PERFORMANCE\_ASSURANCE:
- PREVENTION:
- UNKNOWN:

## **action\_mode**

- AUTOMATIC:
- DISABLED:
- EXTERNAL\_APPROVAL:
- MANUAL:
- RECOMMEND:

## **action\_state**

- ACCEPTED:
- CLEARED:
- FAILED:
- FAILING:
- IN\_PROGRESS:
- POST\_IN\_PROGRESS:
- PRE\_IN\_PROGRESS:
- QUEUED:
- READY:
- REJECTED:
- SUCCEEDED:

## **action\_type**

- ACTIVATE:
- ALLOCATE:
- BUY\_RI:
- DEACTIVATE:
- DELETE:
- MOVE:
- NONE:
- PROVISION:
- RECONFIGURE:
- RESIZE:
- SCALE:
- START:
- SUSPEND:

## **attr\_type**

- ENTITY\_STATE:
- NUM\_VCPU:
- VOLUME\_ATTACHED:

## **cost\_category**

- IP:
- ON\_DEMAND\_COMPUTE:
- ON\_DEMAND\_LICENSE:
- RESERVED\_LICENSE:
- RI\_COMPUTE:
- SPOT:
- STORAGE:
- TOTAL:
- CLOUD\_COMMITMENT\_USAGE:

## **cost\_source**

- BUY\_RI\_DISCOUNT:
- ENTITY\_UPTIME\_DISCOUNT:
- ON\_DEMAND\_RATE:
- RI\_INVENTORY\_DISCOUNT:
- TOTAL:
- UNCLASSIFIED:
- CLOUD\_COMMITMENT\_DISCOUNT:

## **entity\_state**

- FAILOVER:
- MAINTENANCE:
- POWERED\_OFF:
- POWERED\_ON:
- SUSPENDED:
- UNKNOWN:

## **entity\_type**

- ACTION\_MANAGER:
- APPLICATION:
- APPLICATION\_COMPONENT:
- APPLICATION\_SERVER:
- AVAILABILITY\_ZONE:
- BILLING\_FAMILY:
- BUSINESS:
- BUSINESS\_ACCOUNT:
- BUSINESS\_APPLICATION:



- BUSINESS\_ENTITY:
- BUSINESS\_TRANSACTION:
- BUSINESS\_USER:
- CHASSIS:
- CLOUD\_COMMITMENT:
- CLOUD\_SERVICE:
- COMPUTE\_CLUSTER:
- COMPUTE\_RESOURCE:
- COMPUTE\_TIER:
- CONTAINER:
- CONTAINER\_PLATFORM\_CLUSTER:
- CONTAINER\_POD:
- CONTAINER\_SPEC:
- DATABASE:
- DATABASE\_SERVER:
- DATABASE\_SERVER\_TIER:
- DATABASE\_TIER:
- DATACENTER:
- DESIRED\_RESERVED\_INSTANCE:
- DESKTOP\_POOL:
- DISK\_ARRAY:
- DISTRIBUTED\_VIRTUAL\_PORTGROUP:
- DPOD:
- GROUP:
- HCI\_PHYSICAL\_MACHINE:
- HYPERVISOR\_SERVER:
- INFRASTRUCTURE:
- INTERNET:
- IO\_MODULE:
- IP:
- K8S\_CLUSTER:
- LICENSING\_SERVICE:
- LOAD\_BALANCER:
- LOGICAL\_POOL:
- MAC:
- MOVER:
- NAMESPACE:
- NETWORK:
- NETWORKING\_ENDPOINT:
- NODE\_POOL:
- OPERATOR:
- PHYSICAL\_MACHINE:
- PORT:

- PROCESSOR\_POOL:
- REGION:
- RESERVED\_INSTANCE:
- RESERVED\_INSTANCE\_SPECIFICATION:
- RESOURCE\_GROUP:
- RIGHT\_SIZER:
- SAVINGS:
- SERVICE:
- SERVICE\_ENTITY\_TEMPLATE:
- SERVICE\_PROVIDER:
- STORAGE:
- STORAGE\_CLUSTER:
- STORAGE\_CONTROLLER:
- STORAGE\_TIER:
- STORAGE\_VOLUME:
- SWITCH:
- TARGET:
- THIS\_ENTITY:
- THREE\_TIER\_APPLICATION:
- VIEW\_POD:
- VIRTUAL\_APPLICATION:
- VIRTUAL\_DATACENTER:
- VIRTUAL\_MACHINE:
- VIRTUAL\_VOLUME:
- VLAN:
- VM\_SPEC:
- VPOD:
- WEB\_SERVER:
- WORKLOAD\_CONTROLLER:
- \_NONE\_:
- APPLICATION\_COMPONENT\_SPEC:
- VIRTUAL\_MACHINE\_SPEC:

## **environment\_type**

- CLOUD:
- HYBRID:
- ON\_PREM:
- UNKNOWN\_ENV:

## **file\_type**

- CONFIGURATION:
- DISK:
- ESXCONSOLE:

- ISO:
- LOG:
- MEMORY:
- SNAPSHOT:
- SWAP:

## **metric\_type**

- ACTIVE\_SESSIONS:  
Count of the number of actively logged in sessions  
Units: Count
- BALLOONING:  
Number of kb the host is dedicated for ballooning  
Units: kilobytes
- BUFFER\_COMMODITY:  
Utilization of a port buffer (fabric, UCS specific)  
Units: megabytes
- CONNECTION:  
Number of connections onto the application  
Units: Count
- COOLING:
- CPU:  
For Physical machine - usage of CPU (current column). For VM - consumed CPU on the host (consumed column).  
Units: MHz
- CPU\_ALLOCATION:  
For VMs (consumed column) - VM buys from VDC or from desktop pool (VDI). For PMs - PMs sells to VDCs.  
Units: MHz
- CPU\_HEADROOM:  
Number of VMs that can fit in all of the available CPU (on a cluster)  
Units: Count
- CPU\_PROVISIONED:  
Capacity is based on the over-provisioned value in policy  
Units: MHz
- CPU\_READY:  
Ready queue  
Units: ms
- DB\_CACHE\_HIT\_RATE:  
The percentage of accesses that result in cache hits.  
Units: Percentage
- DB\_MEM:

- DB mem utilization or DM capacity  
Units: kilobytes
- DTU:  
Capacity defined in the cloud  
Units: Count
- ENERGY:  
Amount of energy consumed by hosts and virtual machines in an hour.  
Units: watt hours
- EXTENT:  
Internal  
Units: NA
- FLOW:  
Amount of network traffic per second between two nodes  
Units: kilobytes/sec
- FLOW\_ALLOCATION:  
Units: bytes
- HEAP:  
Amount of heap configured for an application  
Units: kilobytes
- IMAGE\_CPU:  
For VDI, usage of the business user  
Units: MHz
- IMAGE\_MEM:  
For VDI, usage of the business user  
Units: kilobytes
- IMAGE\_STORAGE:  
For VDI, usage of the business user  
Units: megabytes
- IO\_THROUGHPUT:  
Amount of traffic through the IO adapter  
Units: kilobytes/sec
- MEM:  
Physical machine - usage of memory. VM - consumed memory.  
Units: kilobytes
- MEM\_ALLOCATION:  
Physical machine - what the host provides to VDCs. VM (consumed column) - what the VM buys from VDC or desktop pool.  
Units: kilobytes
- MEM\_HEADROOM:  
Number of VMs that can fit in the available memory on the cluster

- Units: Count
- MEM\_PROVISIONED:
 

Capacity, based on the over-provisioned value in policy

Units: kilobytes
  - NET\_THROUGHPUT:
 

Amount of traffic through the Network adapter

Units: kilobytes/sec
  - POOL\_CPU:
 

For VDI, usage of the desktop pool

Units: MHz
  - POOL\_MEM:
 

For VDI, usage of the desktop pool

Units: kilobytes
  - POOL\_STORAGE:
 

For VDI, usage of the desktop pool

Units: megabytes
  - PORT\_CHANEL:
 

Traffic on ports for UCS Mgr/HPE1

Units: kilobytes/sec
  - POWER:
 

Rate of power consumption by hosts and virtual machines.

Units: watts
  - Q16\_VCPU:
 

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms
  - Q1\_VCPU:
 

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms
  - Q2\_VCPU:
 

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms
  - Q32\_VCPU:
 

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms
  - Q3\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q4\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q5\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q64\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q6\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q7\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q8\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- QN\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- REMAINING\_GC\_CAPACITY:

Percentage of time not spent on garbage collection

Units: Percentage

- RESPONSE\_TIME:

Application response

Units: ms

- SLA\_COMMODITY:  
User-defined value  
Units: Number
- STORAGE\_ACCESS:  
Count of the number of I/O (read + write) within two discovery cycles  
Units: count
- STORAGE\_ALLOCATION:  
Allocated storage  
Units: megabytes
- STORAGE\_AMOUNT:  
Used storage  
Units: megabytes
- STORAGE\_HEADROOM:  
Count of VM that can fit for the storage in a cluster in terms of storage  
Units: Count
- STORAGE\_LATENCY:  
Acceptable latency. User can set capacity for most storage, is discovered for NetApp.  
Units: ms
- STORAGE\_PROVISIONED:  
Capacity, based on the over-provisioned value in policy  
Units: megabytes
- SWAPPING:  
Memory swapping  
Units: bytes/sec
- THREADS:  
Threads in use  
Units: Count
- TOTAL\_HEADROOM:  
Min value of CPU, Mem, IO, of the headrooms  
Units: Count
- TOTAL\_SESSIONS:  
Total number of possible sessions  
Units: Count
- TRANSACTION:  
Transactions for the application  
Units: transactions/sec
- TRANSACTION\_LOG:  
Size of the transaction log  
Units: megabytes
- VCPU:

- CPU for the VM
- Units: MHz
- VCPU\_LIMIT\_QUOTA:
  - Units: millicores
- VCPU\_REQUEST:
  - Units: millicores
- VCPU\_REQUEST\_QUOTA:
  - Units: millicores
- VMEM:
  - Memory sold by a VM. Memory sold and bought for a container
  - Units: kilobytes
- VMEM\_LIMIT\_QUOTA:
  - Units: kilobytes
- VMEM\_REQUEST:
  - Units: kilobytes
- VMEM\_REQUEST\_QUOTA:
  - Units: kilobytes
- VSTORAGE:
  - Units: megabytes
- ENERGY:

## **savings\_type**

- MISSED\_INVESTMENTS:
  - Actions you could have taken that would increase your investment to provide more resources to your applications.
- MISSED\_SAVINGS:
  - Actions you could have taken that could have reduced your costs.
- REALIZED\_INVESTMENTS:
  - Actions you have taken that increase your investment to provide more resources to your applications.
- REALIZED\_SAVINGS:
  - Actions you have taken that have reduced your costs.

## **severity**

- CRITICAL:
- MAJOR:
- MINOR:
- NORMAL:

## **terminal\_state**

- FAILED:
- SUCCEEDED:



## Data Exporter Objects

The data exporter generates the following JSON objects.

### AccountExpenses

| JSON Object     | Description | Type                       | Repeated | MapKeyType | Reference                  |
|-----------------|-------------|----------------------------|----------|------------|----------------------------|
| expenseDate     |             | String                     | False    |            |                            |
| serviceExpenses |             | <a href="#">CostAmount</a> | False    | String     | <a href="#">CostAmount</a> |
| serviceList     |             | String                     | True     |            |                            |

### Action

| JSON Object  | Description                                                                   | Type                            | Repeated | Format                   | MapKeyType | Reference                       |
|--------------|-------------------------------------------------------------------------------|---------------------------------|----------|--------------------------|------------|---------------------------------|
| buyRiInfo    | Data that analysis uses to calculate Buy RI actions.                          | <a href="#">BuyRiInfo</a>       | False    |                          |            | <a href="#">BuyRiInfo</a>       |
| category     | The category of action.                                                       | <a href="#">action_category</a> | False    |                          |            | <a href="#">action_category</a> |
| creationTime | When analysis created the action.                                             | String                          | False    | yyyy-MM-dd'T'HH:mm:ss'Z' |            |                                 |
| deleteInfo   | A description of a file is unattached and should be deleted.                  | <a href="#">DeleteInfo</a>      | False    |                          |            | <a href="#">DeleteInfo</a>      |
| description  | The description that analysis generates for the action.                       | String                          | False    |                          |            |                                 |
| explanation  | The condition that the action is addressing. For example, Q2 VCPU CONGESTION. | String                          | False    |                          |            |                                 |

| JSON Object | Description                                                           | Type                                 | Repeated | Format | MapKeyType                  | Reference                            |
|-------------|-----------------------------------------------------------------------|--------------------------------------|----------|--------|-----------------------------|--------------------------------------|
| mode        | The mode for the action execution.                                    | <a href="#">action_mode</a>          | False    |        |                             | <a href="#">action_mode</a>          |
| moveInfo    | Information to describe a MOVE action.                                | <a href="#">MoveChange</a>           | False    |        |                             | <a href="#">MoveChange</a>           |
| oid         | The action object identifier.                                         | Long                                 | False    |        |                             |                                      |
| related     | An array of entities or groups that are related to the target entity. | <a href="#">RelatedEntity</a>        | True     |        | <a href="#">entity_type</a> | <a href="#">RelatedEntity</a>        |
| resizeInfo  | Information to describe a RESIZE action.                              | <a href="#">CommodityChange</a>      | False    |        |                             | <a href="#">CommodityChange</a>      |
| savings     | The cost change for the action.                                       | <a href="#">CostAmount</a>           | False    |        |                             | <a href="#">CostAmount</a>           |
| scaleInfo   | Information to describe a SCALE action.                               | <a href="#">MoveChange</a>           | False    |        |                             | <a href="#">MoveChange</a>           |
| severity    | For a given action, the severity of the issue that it addresses.      | <a href="#">severity</a>             | False    |        |                             | <a href="#">severity</a>             |
| state       | The different states for the action, such as completed or pending.    | <a href="#">action_state</a>         | False    |        |                             | <a href="#">action_state</a>         |
| target      | The entity that the action operates on.                               | <a href="#">ActionImpactedEntity</a> | False    |        |                             | <a href="#">ActionImpactedEntity</a> |
| type        | The action type.                                                      | <a href="#">action_type</a>          | False    |        |                             | <a href="#">action_type</a>          |

## Commodity

| JSON Object | Description | Type   |
|-------------|-------------|--------|
| capacity    |             | Double |
| consumed    |             | Double |
| current     |             | Double |
| utilization |             | Double |

## CostAmount

| JSON Object | Description | Type   |
|-------------|-------------|--------|
| amount      |             | Double |
| unit        |             | String |

## CostByCategory

| JSON Object | Description | Type  | MapKeyType         |
|-------------|-------------|-------|--------------------|
| source      |             | Float | <u>cost_source</u> |
| total       |             | Float |                    |

## Entity

| JSON Object     | Description                                                                                              | Type                    | Repeated | MapKeyType         | Reference               |
|-----------------|----------------------------------------------------------------------------------------------------------|-------------------------|----------|--------------------|-------------------------|
| accountExpenses |                                                                                                          | <u>AccountExpenses</u>  | False    |                    | <u>AccountExpenses</u>  |
| attrs           | Additional descriptions for a given entity. The values of this are different for different entity types. | <u>entity.attrs</u>     | False    |                    | <u>entity.attrs</u>     |
| cost            |                                                                                                          | <u>EntityCost</u>       | False    |                    | <u>EntityCost</u>       |
| environment     | The environment type - Hybrid, Cloud, On-prem, etc.                                                      | <u>environment_type</u> | False    |                    | <u>environment_type</u> |
| metric          | For the given metric type, the utilization and capacity of the commodity.                                | <u>Commodity</u>        | False    | <u>metric_type</u> | <u>Commodity</u>        |

| JSON Object | Description                                                                         | Type                          | Repeated                    | MapKeyType | Reference                     |
|-------------|-------------------------------------------------------------------------------------|-------------------------------|-----------------------------|------------|-------------------------------|
| name        | The entity name.                                                                    | String                        | False                       |            |                               |
| oid         | The entity ID.                                                                      | Long                          | False                       |            |                               |
| related     | For the current entity, another entity that is connected to it in the supply chain. | <a href="#">RelatedEntity</a> | <a href="#">entity_type</a> | True       | <a href="#">RelatedEntity</a> |
| state       | The current state of the entity - POWERED_ON, SUSPENDED, MAINTENANCE, etc.          | <a href="#">entity_state</a>  | False                       |            | <a href="#">entity_state</a>  |
| type        | The entity type.                                                                    | <a href="#">entity_type</a>   | False                       |            | <a href="#">entity_type</a>   |

## EntityCost

| JSON Object | Description | Type                           | MapKeyType                    | Reference                      |
|-------------|-------------|--------------------------------|-------------------------------|--------------------------------|
| category    |             | <a href="#">CostByCategory</a> | <a href="#">cost_category</a> | <a href="#">CostByCategory</a> |
| total       |             | Float                          |                               |                                |
| unit        |             | String                         |                               |                                |

## Group

| JSON Object | Description | Type                        | Reference                   |
|-------------|-------------|-----------------------------|-----------------------------|
| attrs       |             | <a href="#">group.attrs</a> | <a href="#">group.attrs</a> |
| name        |             | String                      |                             |
| oid         |             | Long                        |                             |
| type        |             | <a href="#">entity_type</a> | <a href="#">entity_type</a> |

## RelatedEntity

| JSON Object | Description | Type   |
|-------------|-------------|--------|
| name        |             | String |
| oid         |             | Long   |

## Shared Data Objects

The data exporter generates the following JSON objects. Some of these objects are shared with other data management components in Turbonomic.

### ActionCommodity

| Data Object           | Description                                                                           | Type   |
|-----------------------|---------------------------------------------------------------------------------------|--------|
| capacity              | The available capacity of the given commodity.                                        | float  |
| percentileUtilization | For entities and resources that support it, the calculated percentile of utilization. | Double |
| used                  | The amount of the resource currently in use.                                          | float  |
| utilization           | The percentage of capacity in use.                                                    | float  |

### ActionEntity

| Data Object | Description                                                                        | Type               | Reference          |
|-------------|------------------------------------------------------------------------------------|--------------------|--------------------|
| oid         | The ID of the entity.                                                              | Long               |                    |
| name        | The entity name. This corresponds to the name that displays in the user interface. | String             |                    |
| type        | The entity type.                                                                   | <u>entity_type</u> | <u>entity_type</u> |

### ActionImpactedCost

| Data Object          | Description                              | Type  |
|----------------------|------------------------------------------|-------|
| onDemandCost         | For on-demand billing, the entity cost.  | Float |
| onDemandRate         | For on-demand billing, the rate of cost. | Float |
| riCoveragePercentage | For RIs, the percentage of RI coverage.  | Float |

### ActionImpactedCosts

| Data Object   | Description                                                       | Type                      | Reference                 |
|---------------|-------------------------------------------------------------------|---------------------------|---------------------------|
| afterActions  | A description of what the cost will be after the action executes. | <u>ActionImpactedCost</u> | <u>ActionImpactedCost</u> |
| beforeActions | A description of the current cost, before the action executes.    | <u>ActionImpactedCost</u> | <u>ActionImpactedCost</u> |

## ActionImpactedEntity

| Data Object     | Description                                                                                                   | Type                                | MapKeyType                  | Reference                           |
|-----------------|---------------------------------------------------------------------------------------------------------------|-------------------------------------|-----------------------------|-------------------------------------|
| affectedCosts   | A description of the before and after costs for this entity.                                                  | <a href="#">ActionImpactedCosts</a> |                             | <a href="#">ActionImpactedCosts</a> |
| affectedMetrics | A description of the before and after metrics for this entity.                                                | <a href="#">ImpactedMetric</a>      | <a href="#">metric_type</a> | <a href="#">ImpactedMetric</a>      |
| name            | The entity name.                                                                                              | String                              |                             |                                     |
| oid             | The entity ID.                                                                                                | Long                                |                             |                                     |
| settings        | For entities that support percentile calculations, the aggressiveness and observation period for this entity. | <a href="#">EntitySettings</a>      |                             | <a href="#">EntitySettings</a>      |
| type            | The entity type.                                                                                              | <a href="#">entity_type</a>         |                             | <a href="#">entity_type</a>         |

## BuyRiInfo

| Data Object   | Description                                                                                                              | Type                         | Reference                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------|
| computeTier   | The compute tier, or workload family for the action. The ActionEntity object will be for an entity of type COMPUTE_TIER. | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| count         |                                                                                                                          | Integer                      |                              |
| masterAccount | The account that manages the RI billing.                                                                                 | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| region        | The region that will host the RIs.                                                                                       | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| target        | The cloud account or subscription that will manage the RIs.                                                              | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |

## CommodityChange

| Data Object   | Description                                                             | Type                        | Reference                   |
|---------------|-------------------------------------------------------------------------|-----------------------------|-----------------------------|
| attribute     |                                                                         | String                      |                             |
| commodityType | The type of commodity the action will change - MEM, IO_THROUGHPUT, etc. | <a href="#">metric_type</a> | <a href="#">metric_type</a> |
| from          | The commodity value before the action executes.                         | Float                       |                             |

| Data Object      | Description                                                                                                                             | Type                                      | Reference                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------|
| percentileChange | For entities and commodities that support percentile calculations, the before and after values for percentile utilization and settings. | <a href="#">CommodityPercentileChange</a> | <a href="#">CommodityPercentileChange</a> |
| target           | A description of the entity this action will change.                                                                                    | <a href="#">ActionEntity</a>              | <a href="#">ActionEntity</a>              |
| to               | The commodity value after the action executes.                                                                                          | Float                                     |                                           |
| unit             | The displayed units of measurement.                                                                                                     | String                                    |                                           |

## CommodityPercentileChange

| Data Object           | Description                                                  | Type    |
|-----------------------|--------------------------------------------------------------|---------|
| after                 | The percentile utilization after the action executes         | Double  |
| aggressiveness        | The aggressiveness setting used to calculate the percentile. | Integer |
| before                | The percentile utilization before the action executes.       | Double  |
| observationPeriodDays | The observation period used to calculate the percentile.     | Integer |

## DeleteInfo

| Data Object           | Description                                      | Type    | Format                   |
|-----------------------|--------------------------------------------------|---------|--------------------------|
| filePath              | The path to the file.                            | String  |                          |
| fileSize              | The size of the file.                            | Double  |                          |
| lastModifiedTimestamp | When the file was last modified.                 | String  | yyyy-MM-dd'T'HH:mm:ss'Z' |
| unattachedDays        | How many days the file has been unattached.      | Integer |                          |
| unit                  | The displayed unit of measure for the file size. | String  |                          |

## entity.attrs

| Data Object        | Description                                              | Type                                                        | SupportedEntityTypes | Repeated | Reference              |
|--------------------|----------------------------------------------------------|-------------------------------------------------------------|----------------------|----------|------------------------|
| attachment_state   | Whether a given volume is attached or unattached.        | AttachmentState                                             | VIRTUAL_VOLUME       | False    | AttachmentState        |
| connected_networks | A list of network names that a given VM is connected to. | MULTI_TEXT                                                  | VIRTUAL_MACHINE      | True     |                        |
| cpu_model          | For a physical host, the model of CPU on that host.      | TEXT                                                        | PHYSICAL_MACHINE     | False    |                        |
| guest_os_name      | The name of the OS running on a given VM.                | TEXT                                                        | VIRTUAL_MACHINE      | False    |                        |
| guest_os_type      | The specific type for the OS running on a given VM.      | <a href="#">OSType</a>                                      | VIRTUAL_MACHINE      | False    | <a href="#">OSType</a> |
| is_local           | Whether a storage is local to the machine, or networked. | BOOLEAN                                                     | STORAGE              | False    |                        |
| model              | The specific model of the given physical host.           | TEXT                                                        | PHYSICAL_MACHINE     | False    |                        |
| num_cpus           | The number of CPUs on the given physical host.           | NUMBER                                                      | PHYSICAL_MACHINE     | False    |                        |
| tags               | A list of tags that are assigned to the given entity.    | TEXT_MULTIMAPP (Embedded Reporting) or List (Data Exporter) | ALL                  | False    |                        |
| targets            | The target that manages the given entity.                | <a href="#">Target</a>                                      | ALL                  | True     | <a href="#">Target</a> |



| Data Object | Description                                                     | Type | SupportedEntityTypes            | Repeated  | Reference |
|-------------|-----------------------------------------------------------------|------|---------------------------------|-----------|-----------|
| timezone    | The timezone that is set for the clock on the physical machine. | TEXT | PHYSICAL_MACHINE                | False     |           |
| vendor_id   |                                                                 | TEXT | VIRTUAL_VOLUME,BUSINESS_ACCOUNT | Reference |           |

## EntitySettings

| Data Object                     | Description                         | Type    |
|---------------------------------|-------------------------------------|---------|
| percentileAggressiveness        | The current aggressiveness setting. | Integer |
| percentileObservationPeriodDays | The current observation period.     | Integer |

## group.attrs

| Data Object  | Description                                                | Type       | Repeated | Reference | SupportedEntityTypes                                                            | SupportedGroupTypes                                                             |
|--------------|------------------------------------------------------------|------------|----------|-----------|---------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| dynamic      | Whether this group is static or dynamic. true for dynamic. | BOOLEAN    | False    |           | STORAGE_CLUSTER,COMPUTE_CLUSTER,GROUP,K8S_CLUSTER,BILLING_FAMILY,RESOURCE_GROUP | RESOURCE_GROUP,COMPUTE_CLUSTER,BILLING_FAMILY,GROUP,K8S_CLUSTER,STORAGE_CLUSTER |
| member_types |                                                            | MULTI_TEXT | True     |           | STORAGE_CLUSTER,COMPUTE_CLUSTER,GROUP,K8S_CLUSTER,BILLING_FAMILY,RESOURCE_GROUP | RESOURCE_GROUP,COMPUTE_CLUSTER,BILLING_FAMILY,GROUP,K8S_CLUSTER,STORAGE_CLUSTER |
| origin       |                                                            | TEXT       | False    |           | STORAGE_CLUSTER,COMPUTE_CLUSTER,GROUP,K8S_CLUSTER,BILLING_FAMILY,RESOURCE_GROUP | RESOURCE_GROUP,COMPUTE_CLUSTER,BILLING_FAMILY,GROUP,K8S_CLUSTER,STORAGE_CLUSTER |

| Data Object | Description | Type                                                              | Repeated | Reference     | SupportedEntityTypes                                                            | SupportedGroupTypes |
|-------------|-------------|-------------------------------------------------------------------|----------|---------------|---------------------------------------------------------------------------------|---------------------|
| tags        |             | TEXT_MULTI<br>MAP<br>(Embedded Reporting) or List (Data Exporter) | False    |               | GROUP,RESOURCE_GROUP,COMPUTE_CLUSTER,K8S_CLUSTER,STORAGE_CLUSTER,BILLING_FAMILY | ALL                 |
| targets     |             | <u>Target</u>                                                     | False    | <u>Target</u> | GROUP,RESOURCE_GROUP,COMPUTE_CLUSTER,K8S_CLUSTER,STORAGE_CLUSTER,BILLING_FAMILY | ALL                 |

## ImpactedMetric

| Data Object   | Description                                                                     | Type                   | Reference              |
|---------------|---------------------------------------------------------------------------------|------------------------|------------------------|
| afterActions  | A description of what the metric utilization will be after the action executes. | <u>ActionCommodity</u> | <u>ActionCommodity</u> |
| beforeActions | A description of the current utilization, before the action executes.           | <u>ActionCommodity</u> | <u>ActionCommodity</u> |

## MoveChange

| Data Object | Description                      | Type                        | Repeated | Reference                   |
|-------------|----------------------------------|-----------------------------|----------|-----------------------------|
| from        | Where the entity will move from. | <u>ActionImpactedEntity</u> | False    | <u>ActionImpactedEntity</u> |
| resource    | The entity that will move.       | <u>ActionEntity</u>         | True     | <u>ActionEntity</u>         |
| to          | Where the entity will move to.   | <u>ActionImpactedEntity</u> | False    | <u>ActionImpactedEntity</u> |

## Target

| Data Object    | Description | Type   |
|----------------|-------------|--------|
| category       |             | String |
| entityVendorId |             | String |
| name           |             | String |
| oid            |             | Long   |

| Data Object | Description | Type   |
|-------------|-------------|--------|
| type        |             | String |

## action.attrs

| JSON Object | Description                                                               | Type                                 | MapKeyType                  | Reference                            |
|-------------|---------------------------------------------------------------------------|--------------------------------------|-----------------------------|--------------------------------------|
| buyRiInfo   | For an action to buy RIs, additional information about the RIs.           | <a href="#">BuyRiInfo</a>            |                             | <a href="#">BuyRiInfo</a>            |
| deleteInfo  | For an action to delete a file, a description of that file.               | <a href="#">DeleteInfo</a>           |                             | <a href="#">DeleteInfo</a>           |
| moveInfo    | For a move action, a description of from and to for the move.             | <a href="#">MoveChange</a>           | <a href="#">entity_type</a> | <a href="#">MoveChange</a>           |
| resizeInfo  | For a resize action, a description of the commodity changes.              | <a href="#">CommodityChange</a>      | <a href="#">metric_type</a> | <a href="#">CommodityChange</a>      |
| scaleInfo   | For scaling a cloud VM, a description of the from and to for the VM move. | <a href="#">MoveChange</a>           | <a href="#">entity_type</a> | <a href="#">MoveChange</a>           |
| target      | Extra information about the entity this action impacts.                   | <a href="#">ActionImpactedEntity</a> |                             | <a href="#">ActionImpactedEntity</a> |

## Reference: Working with YAML Files

**YAML** is the primary file format to create and configure resources on kubernetes, including everything to do with the Turbonomic platform. The Custom Resource YAML provides a convenient, single place to define the majority of configuration details for Turbonomic. When editing YAML files, you must be careful to respect the file syntax, including indents in the file. General rules for edits include:

- Always uses spaces, not tabs, for all indentation. Since spacing and indentations matter, and can yield an invalid result or a parameter completely skipped over, you should work with YAML files with an editor that supports using vertical lines associated with indentations to visually spot a misalignment. If your editor of choice makes this difficult, you can use the linux expand utility when you're done, to convert tabs to equivalent spaces.
- Indention uses two spaces per level.
- Be careful to keep the same indentation for all properties in a given section.
- Never use the same property name twice in the same section. Doing this will render the YAML file invalid, though you will likely not see any notification of a problem. Rather, all but one of the property definitions will be silently ignored.

### Spacing matters

For the Turbonomic Custom Resource, indentation defines where parameters are applied (globally or to specific components), so ensure you line up the text appropriately. The following example shows examples of specifications that are applied at a global level (`{ "spec": { "global":`

[{"tag": "8.6.4"}]})) which sets the container image tag for all instances. Then indented we see properties that are global for the remote database ({"spec": {"properties": {"global": [{"dbPort": "6033"}]}]})) describes a property of dbPort that would be set for remote DB connections.

Each line is indented two spaces from the higher level.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

## Combine properties correctly

A YAML file is read top down, and if there are different parameters that apply to the same component, they need to be combined. The following example shows a YAML where properties for the ui component of image tag *and* memory limit resources.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
 ui:
 resources:
 limits:
 memory: 4Gi
```

This YAML will not set both image tag and memory limit resources for the ui component because they are set in two different sections. The memory limits would be applied since that is the last section to be read, and it would overwrite the first set of properties with the image tag. The following YAML will set both image tag and memory limit resources for the ui component:

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 resources:
 limits:
 memory: 4Gi
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

To avoid having the same property name twice in the same section, search for an existing instance of the key you want to add. With the complex embedded structure in many k8s resources, this can be quite challenging; however, you can use an online validator that does not accept duplicate keys to check your entire CR file (such as [YAMLint - The YAML Validator](#)).

## Finding a property

In this documentation we refer to specific properties in the CR file using a "path" expression. For example, assume the path `/spec/global/repository` to designate a `repository` property in the file. You can find the property in the file as follows:

1. Find a line that says `spec :` with no indentation at all.
2. Between that line and the next unindented line (not counting comments, which start with `#`), find a line that says `global :` and is at the next level of indentation.
3. Between that line and the next line with the same indentation, find a line that starts with `repository :`. That line is where the addressed property is defined.

In this example, find the `repository` property specified by `/spec/global/repository`:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: Xl
metadata:
 name: xl-release
spec:
 properties:
 global:
 repository: # This is NOT the correct property
 ...

 # Global settings
 global:
 repository: # This is the one we're after
```

The first `repository` property is not at `/spec/global/repository`, but at `/spec/properties/global/repository`.



---

## Chapter 12. Integration - Embedded Reporting

The Turbonomic platform includes an Embedded Reporting component that you can choose to enable when you install the platform. Embedded Reporting stores a history of your managed environment and then presents selective snapshots of this history via a set of standard dashboards and reports.

Embedded Reporting runs as its own component, as part of the Turbonomic platform. This architecture enhances performance and reduces storage requirements. It stores a history of your managed environment and then presents selective snapshots of this history via a set of standard dashboards and reports. You can create your own dashboards and reports to focus on other areas of concern.

Dashboards and charts in Turbonomic Embedded Reporting are powered by the Grafana® observability platform. With Grafana, it's easy to navigate the existing dashboards, and to make your own charts and dashboards with no coding required. You can also create custom reports via SQL queries against the Embedded Reporting database.

To compose custom reports, open the user interface, navigate to the Reports view, and then explore the Timescale database. From there, you can assemble queries to populate your custom reports. To help you build these queries, this documentation describes the schema tables, and provides some sample queries against them.

The Embedded Reporting feature uses a TimescaleDB service to manage the reporting data. The default installation of Turbonomic includes its own installation of TimescaleDB. This documentation also includes instructions to deploy an external TimescaleDB service. You can use this as an alternative to the default deployment.

For instructions to enable Embedded Reporting, see [Enabling Embedded Reporting](#).

---

### Custom Reports

To get the most out of Embedded Reports, you will display custom reports on the Grafana dashboard page. You can import reports from the Turbonomic Open Source site, or you can create your own using SQL Server queries into the data sets.

### Importing Reports

You can find a set of custom reports to import on the Turbonomic Open Source site at <https://github.com/turbonomic/visualization/tree/main/embedded-visualization>. This page lists different categories of reports that you can use to track your environment.

To import a custom report:

1. In the Embedded Reports page, create a folder to store your imported JSON files.

In Embedded Reports page, navigate to **Dashboards / Browse**. on that page, click **New Folder** to create a folder for your imported JSON files.

For example, create a folder named `{{ site.data.vars.Product_Short }}_Github_Reports`.

After you create the folder, navigate back to the **Dashboards / Browse** page.

2. Navigate to the report you want to import.

Go to <https://github.com/turbonomic/visualization/tree/main/embedded-visualization> and browse to find the report you want.

3. Copy the JSON file to the clipboard.

In the report entry, navigate to the JSON file. In GitHub, display the file as **Raw JSON**, then select the JSON and copy it to the clipboard.

4. Import the JSON file into your Reports.

In the Embedded Reports page, click **Import**, and then paste the JSON you copied into the **Import Via Panel JSON** field. Then click **Load**.

To complete the import:

- Give the name you want for the report.
- Choose the folder that you just created to store the report.
- If you want to share this report with other Grafana installations, make a note of the UID, or change it to a value you can remember.
- Click **Import**.

The Embedded Reports page now displays your imported report.

## Creating Reports

To compose custom reports, your Turbonomic user account must have **Report Editor** permission. When you navigate from Turbonomic to the Grafana Reports page, in addition to browsing dashboards, you can create new dashboards. To create a custom report:

1. Create a new dashboard.

Navigate to the Grafana Dashboards page, then click the **+** icon in the left-hand menu bar to create a new dashboard.

2. Click to create a new panel.

The page for the new panel includes a tabbed section with a **Query** tab.

3. In the **Query** tab, set the **Data Source** field to *Turbo Timescale*.

This sets up the panel to display data from the `{{ site.data.vars.Product_Short }}` Timescale database.

4. Edit the panel's query.

The panel will display a default query. You can edit this query to change the data the panel displays. At the bottom of the default query, click **Edit SQL**. This displays a field where you can enter the SQL that you want.

For information about experimenting with different queries, see [Exploring SQL Queries](#)

To help you build these queries, this document describes the schema tables, and provides some sample queries against them.

## Exploring SQL Queries

To work with SQL queries, your `{{ site.data.vars.Product_Short }}` user account must have **Report Editor** permission. To explore queries:

1. Navigate to the Explore page.

The page displays with a sample query already in place

2. Choose to explore Turbo Timescale.

Next to the page title, **Explore**, you can choose which database to work with. Choose the database *Turbo Timescale*.

If you have installed an external TimescaleDB, be sure to choose the name that you gave to that database.

3. Edit the **Explore** query.

The page displays a default query. You can edit this query to change the data the page displays. At the bottom of the default query, click **Edit SQL**. This displays a field where you can enter the SQL that you want.

To test your query, click **Run Query**. If your query includes data points over time, you can choose **Format As: Time Series**. For tabular data, choose **Format As: Table**.



To help you build these queries, this document describes the schema tables, and provides some sample queries against them.

## Schema Tables

The Embedded Reports data set includes the following Schema Tables to represent Turbonomic data.

### action\_group Table

| Data Object | Type   | Nullable | Primary | Description                                                      | Reference                       |
|-------------|--------|----------|---------|------------------------------------------------------------------|---------------------------------|
| id          | int    | False    | False   |                                                                  |                                 |
| type        | enum   | False    | False   | The specific action type - Provision, Reconfigure, Scale, etc.   | <a href="#">action_type</a>     |
| category    | enum   | False    | False   | The category of the given action.                                | <a href="#">action_category</a> |
| state       | enum   | False    | False   | That action type.                                                | <a href="#">action_state</a>    |
| mode        | enum   | False    | False   | The action mode - automated, manual, etc.                        | <a href="#">action_mode</a>     |
| risks       | text[] | True     | False   | An array of strings to describe the risks that drive the action. |                                 |

### cloud\_service\_cost Table

For a given cloud service that is associated with a cloud target, the cost of that service.

| Data Object       | Type                     | Nullable | Primary | Description                                                                  | Reference |
|-------------------|--------------------------|----------|---------|------------------------------------------------------------------------------|-----------|
| time              | timestamp with time zone | False    | True    |                                                                              |           |
| account_oid       | bigint                   | False    | True    | The ID of the cloud account or subscription this service is associated with. |           |
| cloud_service_oid | bigint                   | False    | True    | The ID of the cloud service.                                                 |           |
| cost              | real                     | False    | False   | The cost of the service.                                                     |           |

### completed\_action Table

This table lists the current set of completed actions. A completed action is an action that completed its execution, whether the action succeeded or failed.

This table does not contain "in progress" or "queued" actions.

| Data Object         | Type                     | Nullable | Primary | Description                           | Reference |
|---------------------|--------------------------|----------|---------|---------------------------------------|-----------|
| recommendation_time | timestamp with time zone | False    | False   | When analysis recommended the action. |           |

| Data Object       | Type                     | Nullable | Primary | Description                                                                                           | Reference                       |
|-------------------|--------------------------|----------|---------|-------------------------------------------------------------------------------------------------------|---------------------------------|
| acceptance_time   | timestamp with time zone | False    | False   | For manual actions, when the action was accepted for execution.                                       |                                 |
| completion_time   | timestamp with time zone | False    | False   | When the action completed. Can be for success or failure.                                             |                                 |
| action_oid        | bigint                   | False    | False   | The ID of the action. Use this to get action details.                                                 |                                 |
| type              | enum                     | False    | False   | The action type.                                                                                      | <a href="#">action_type</a>     |
| severity          | enum                     | False    | False   | The action severity.                                                                                  | <a href="#">severity</a>        |
| category          | enum                     | False    | False   | The action category.                                                                                  | <a href="#">action_category</a> |
| target_entity_id  | bigint                   | False    | False   | The ID of the entity that the action affects. For example, for a VMem resize, the VM that is resized. |                                 |
| involved_entities | bigint[]                 | True     | False   | An array of IDs for entities involved in the action. This includes the target entity.                 |                                 |
| description       | text                     | False    | False   | The text description of the action.                                                                   |                                 |
| savings           | double precision         | False    | False   | The cost savings this action results in. A negative savings indicates an investment.                  |                                 |

| Data Object   | Type  | Nullable | Primary | Description                                                                                                                             | Reference                      |
|---------------|-------|----------|---------|-----------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| final_state   | enum  | False    | False   | Whether the action succeeded or failed.                                                                                                 | <a href="#">terminal_state</a> |
| final_message | text  | False    | False   | A message for the final state. If the action failed, this should contain an error message.                                              |                                |
| attrs         | jsonb | True     | False   | Additional (often type-specific) info for the action. For example, in resizes this will contain the commodity type and from/to amounts. | <a href="#">action.attrs</a>   |

## Sample Use Cases

To get the target entities for all the completed actions, join this table with the entities table. This example lists the completed action type and time, and the entity name type:

```
select completed_action.type AS action_type, completed_action.completion_time AS time,
entity.name, entity.type AS ent_type
FROM completed_action
INNER JOIN entity ON completed_action.target_entity_id=entity.oid
```

## entity Table

The Entity table contains information about which entities appear in the topologies and when. This table gets updated with each market cycle.

| Data Object | Type   | Nullable | Primary | Description      | Reference                   |
|-------------|--------|----------|---------|------------------|-----------------------------|
| oid         | bigint | False    | True    | The entity ID.   |                             |
| type        | enum   | False    | False   | The entity type. | <a href="#">entity_type</a> |

| Data Object | Type                     | Nullable | Primary | Description                                                                                                                     | Reference                                                   |
|-------------|--------------------------|----------|---------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|
| name        | text                     | False    | False   | The current display name for the entity. While display names may change over time, this is always the most recent display name. |                                                             |
| environment | enum                     | True     | False   | Whether the entity is in an on-prem, cloud, or hybrid environment.                                                              | <a href="#">environment_type</a>                            |
| attrs       | jsonb                    | True     | False   | Additional data for the entity.                                                                                                 | <a href="#">entity.attrs</a><br><a href="#">group.attrs</a> |
| first_seen  | timestamp with time zone | False    | False   | When the entity was first discovered in the topology.                                                                           |                                                             |
| last_seen   | timestamp with time zone | False    | False   | When the entity last existed in the topology. For an entity that still exists, the value is 9999-12-31T23:59:59.                |                                                             |

## Sample Use Cases

### Count entities that meet specific criteria:

For a given time interval, count the number of PHYSICAL\_MACHINE entities that have 4 CPUS.

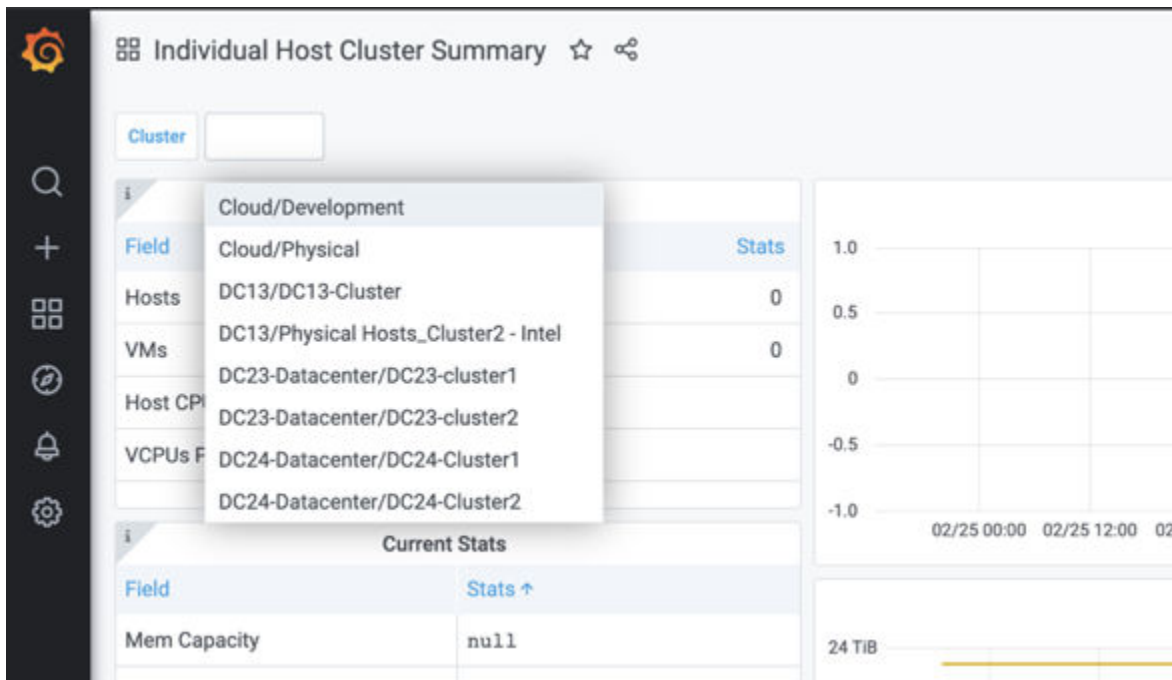
```
select count(*)
from entity e
where e.type = 'PHYSICAL_MACHINE'
 and attrs->'num_cpus' = '4'
 and ('2021-02-01T05:00:00Z', '2021-02-18T04:59:59Z') OVERLAPS (e.first_seen, e.last_seen)
```

### Find clusters that were configured for a given time range:

Queries like this are common to generate values for grafana variables that you can run reports against.

```
SELECT name, oid
FROM entity
WHERE
 entity.type = 'COMPUTE_CLUSTER'
 AND ('2021-02-01T05:00:00Z', '2021-02-18T04:59:59Z') OVERLAPS (first_seen, last_seen)
ORDER BY 1
```

For example, here's a list of variables that a user is choosing:



## entity\_cost Table

For a given cloud entity (a workload such as a VM or a DB service), this table represents a given cost associated with that entity.

| Data Object | Type                     | Nullable | Primary | Description                                                                               | Reference                     |
|-------------|--------------------------|----------|---------|-------------------------------------------------------------------------------------------|-------------------------------|
| time        | timestamp with time zone | False    | False   |                                                                                           |                               |
| entity_oid  | bigint                   | False    | False   | The entity ID.                                                                            |                               |
| category    | enum                     | False    | False   | The category of service cost - IP service, storage service, on-demand or RI compute, etc. | <a href="#">cost_category</a> |
| source      | enum                     | False    | False   | The type of billing data that provides the cost data.                                     | <a href="#">cost_source</a>   |
| cost        | real                     | False    | False   | The calculated cost.                                                                      |                               |

## entity\_savings Table

Turbonomic tracks actions that result in savings or investments in your environment. Actions that add resources usually require an investment, and actions that suspend or reduce resources usually give you a saving. This table describes the cost change of an action in either case.

This table can be in an action whether you have already executed the action or not. The `savings_type` field describes:

- Whether the action is a saving or investment
- Whether the action has been executed or not

| Data Object  | Type                     | Nullable | Primary | Description                                                                                                                                 | Reference                    |
|--------------|--------------------------|----------|---------|---------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| time         | timestamp with time zone | False    | True    | For a realized saving or investment, when the action was executed. For a missed saving or investment, when the action was last recommended. |                              |
| entity_oid   | bigint                   | False    | True    | The object ID for the entity that has seen a saving or investment.                                                                          |                              |
| savings_type | enum                     | False    | True    | Whether the action resulted in an investment or saving, and whether the action executed or not.                                             | <a href="#">savings_type</a> |
| stats_value  | real                     | False    | False   | The value of the investment or saving for the related action.                                                                               |                              |

## entity\_types\_with\_cost Table

This table lists the types of entities that can have costs associated with them. (For costs associated with entities, see [entity\\_cost](#) Table.)

| Data Object | Type | Nullable | Primary | Description      | Reference                   |
|-------------|------|----------|---------|------------------|-----------------------------|
| type        | enum | True     | False   | The entity type. | <a href="#">entity_type</a> |

## file Table

This table stores the paths to files in the environment, as well as properties of the files such as last modification or file size.

| Data Object       | Type                     | Nullable | Primary | Description                                             | Reference                 |
|-------------------|--------------------------|----------|---------|---------------------------------------------------------|---------------------------|
| volume_oid        | bigint                   | False    | True    | If present, the ID of the volume that stores this file. |                           |
| path              | text                     | False    | True    | The path to the file.                                   |                           |
| type              | enum                     | False    | False   | The file type - ISO, log file, snapshot, etc.           | <a href="#">file_type</a> |
| file_size_kb      | bigint                   | True     | False   | The file size in kb.                                    |                           |
| modification_time | timestamp with time zone | True     | False   | When the file was last modified.                        |                           |
| storage_oid       | bigint                   | False    | False   | If present, the ID of the disk that stores this file.   |                           |
| is_attached       | boolean                  | False    | False   | Whether the file is attached or unattached.             |                           |
| hash              | bigint                   | False    | False   |                                                         |                           |

## historical\_entity\_attrs Table

This table tracks the historical attributes associated with an entity. Each row represents the value of a particular attribute of an entity at a particular point in time. New rows are written when the value of a property changes between topologies, and at regular intervals (e.g. daily) to keep a recent value available within a reasonable time range.

| Data Object  | Type                     | Nullable | Primary | Description                                                                                                                                                                                                                                                                              | Reference                 |
|--------------|--------------------------|----------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|
| time         | timestamp with time zone | False    | False   | When this attribute was recorded.                                                                                                                                                                                                                                                        |                           |
| entity_oid   | bigint                   | False    | False   | The ID of the entity this attribute pertains to.                                                                                                                                                                                                                                         |                           |
| type         | enum                     | False    | False   | The attribute type.                                                                                                                                                                                                                                                                      | <a href="#">attr_type</a> |
| bool_value   | boolean                  | True     | False   | If the attribute is boolean, the boolean value.                                                                                                                                                                                                                                          |                           |
| int_value    | int                      | True     | False   | If the attribute is an integer, the integer value. If the attribute is an enum, this is the OID of the enum in the pg_enum table (to guard against name or order changes). The int_value should be castable to the enum via a custom cast (e.g. int_value::entity_state = 'POWERED_ON'). |                           |
| long_value   | bigint                   | True     | False   | If the attribute is a long, the long value.                                                                                                                                                                                                                                              |                           |
| double_value | double precision         | True     | False   | If the attribute is a double, the double value.                                                                                                                                                                                                                                          |                           |

| Data Object      | Type     | Nullable | Primary | Description                                                                                                                                                                              | Reference |
|------------------|----------|----------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| string_value     | text     | True     | False   | If the attribute is a string, the string value.                                                                                                                                          |           |
| int_arr_value    | int[]    | True     | False   | If the attribute is a list of ints, the list values. If the attribute is a list of enums, these are the OIDs of the enums in the pg_enum table (to guard against name or order changes). |           |
| long_arr_value   | bigint[] | True     | False   | If the attribute is a list of longs, the long values.                                                                                                                                    |           |
| string_arr_value | text[]   | True     | False   | If the attribute is a list of strings, the string values.                                                                                                                                |           |
| double_arr_value | float8[] | True     | False   | If the attribute is a list of doubles, the double values.                                                                                                                                |           |
| json_value       | jsonb    | True     | False   | If the attribute is a JSON object, the given object.                                                                                                                                     |           |

## Sample Use Cases

### Get All Powered On Entities:

This query returns all the entities in the historical\_entity\_attrs that do not have a powered\_off state given a certain time range.

```
select *
from historical_entity_attrs h1
where h1.type = 'ENTITY_STATE' and h1.int_value::entity_state != 'POWERED_OFF' AND h1.time
between $__timeFrom() and $__timeTo()
```

### Gap-filling data:

Data in this table are not written at every broadcast. Because of this, if we ask for those data in a time range, we might only get few data points. Consider this query in which we get the values for one entity in a time range:

```
SELECT * FROM historical_entity_attrs h1
where h1.type = 'ENTITY_STATE'
and entity_oid = '73864029152384'
and time between (timestampz '2021-03-01T01:00:00Z' - interval '24 hour') and
```



```
'2021-03-03T04:59:59Z'
order by time;
```

| time                | entity_oid         | type         | bool_value | int_value | long_value | double_value | string_value |
|---------------------|--------------------|--------------|------------|-----------|------------|--------------|--------------|
| 2021-02-27 20:39:50 | 73,864,029,152,384 | ENTITY_STATE | [NULL]     | 17,054    | [NULL]     | [NULL]       | [NULL]       |
| 2021-02-27 21:39:50 | 73,864,029,152,384 | ENTITY_STATE | [NULL]     | 17,056    | [NULL]     | [NULL]       | [NULL]       |
| 2021-02-28 10:59:50 | 73,864,029,152,384 | ENTITY_STATE | [NULL]     | 17,056    | [NULL]     | [NULL]       | [NULL]       |
| 2021-03-01 11:09:50 | 73,864,029,152,384 | ENTITY_STATE | [NULL]     | 17,056    | [NULL]     | [NULL]       | [NULL]       |
| 2021-03-02 11:16:23 | 73,864,029,152,384 | ENTITY_STATE | [NULL]     | 17,056    | [NULL]     | [NULL]       | [NULL]       |

In order to solve this problem we use a timescaledb function that fills the time ranges with data, as if those data were written at a constant interval. This means that if we want data every interval  $t$  but we only have data at  $t_1$  and  $t_4$ , with this function we can generate data at  $t_2$  and  $t_3$  with the same value of  $t_1$ . The function that we use is called `time_bucket_gapfill` and here's a sample query. Notice how in the result we have data points at a constant interval of 10 minutes:

```
SELECT
 time_bucket_gapfill('10 min', time) AS minutes,
 locf(max(h1.int_value::entity_state))
FROM historical_entity_attrs h1
where h1.type = 'ENTITY_STATE'
-- AND h1.int_value::entity_state = 'POWERED_ON'
-- and entity_oid = '73864029152384'
-- and time between (timestampz '2021-03-01T01:00:00Z' - interval '24 hour') and
'2021-03-03T04:59:59Z'
GROUP BY minutes
```

| minutes             | locf        |
|---------------------|-------------|
| 2021-02-27 20:30:00 | POWERED_ON  |
| 2021-02-27 20:40:00 | POWERED_ON  |
| 2021-02-27 20:50:00 | POWERED_ON  |
| 2021-02-27 21:00:00 | POWERED_ON  |
| 2021-02-27 21:10:00 | POWERED_ON  |
| 2021-02-27 21:20:00 | POWERED_ON  |
| 2021-02-27 21:30:00 | POWERED_OFF |
| 2021-02-27 21:40:00 | POWERED_OFF |
| 2021-02-27 21:50:00 | POWERED_OFF |
| 2021-02-27 22:00:00 | POWERED_OFF |
| 2021-02-27 22:10:00 | POWERED_OFF |
| 2021-02-27 22:20:00 | POWERED_OFF |
| 2021-02-27 22:30:00 | POWERED_OFF |
| 2021-02-27 22:40:00 | POWERED_OFF |
| 2021-02-27 22:50:00 | POWERED_OFF |
| 2021-02-27 23:00:00 | POWERED_OFF |

### Casting Entity States to Integers:

Entity States are internally represented as integers. This means that a property such as `POWERED_OFF` is stored as a number. In order to work with this the queries need to cast the integer to the corresponding entity state. This is done with the following cast, supported by Postgres: `int_value::entity_state`. So for example, take a look at the two queries below, the first one without the cast, and the second query with it:

```
SELECT time, entity_oid, entity_state FROM historical_entity_attrs h1
where h1.type = 'ENTITY_STATE'
 and entity_oid = '73864029152384'
 and time between (timestampz '2021-03-01T01:00:00Z' - interval '24 hour') and
'2021-03-03T04:59:59Z'
order by time;
```

| minutes | locf                            |
|---------|---------------------------------|
| 4       | 2021-02-27 20:30:00 POWERED_ON  |
| 5       | 2021-02-27 20:40:00 POWERED_ON  |
| 6       | 2021-02-27 20:50:00 POWERED_ON  |
| 7       | 2021-02-27 21:00:00 POWERED_ON  |
| 8       | 2021-02-27 21:10:00 POWERED_ON  |
| 9       | 2021-02-27 21:20:00 POWERED_ON  |
| 10      | 2021-02-27 21:30:00 POWERED_OFF |
| 11      | 2021-02-27 21:40:00 POWERED_OFF |
| 12      | 2021-02-27 21:50:00 POWERED_OFF |
| 13      | 2021-02-27 22:00:00 POWERED_OFF |
| 14      | 2021-02-27 22:10:00 POWERED_OFF |
| 15      | 2021-02-27 22:20:00 POWERED_OFF |
| 16      | 2021-02-27 22:30:00 POWERED_OFF |
| 17      | 2021-02-27 22:40:00 POWERED_OFF |
| 18      | 2021-02-27 22:50:00 POWERED_OFF |
| 19      | 2021-02-27 23:00:00 POWERED_OFF |

```
SELECT time, entity_oid, int_value::entity_state FROM historical_entity_attrs h1
where h1.type = 'ENTITY_STATE'
 and entity_oid = '73864029152384'
 and time between (timestampz '2021-03-01T01:00:00Z' - interval '24 hour') and
'2021-03-03T04:59:59Z'
order by time;
```

| time | entity_oid          | int_value                      |
|------|---------------------|--------------------------------|
| 1    | 2021-02-27 20:39:50 | 73,864,029,152,384 POWERED_ON  |
| 2    | 2021-02-27 21:39:50 | 73,864,029,152,384 POWERED_OFF |
| 3    | 2021-02-28 10:59:50 | 73,864,029,152,384 POWERED_OFF |
| 4    | 2021-03-01 11:09:50 | 73,864,029,152,384 POWERED_OFF |
| 5    | 2021-03-02 11:16:23 | 73,864,029,152,384 POWERED_OFF |

## metric Table

This table contains metric values for entities that appeared in the topology. Most metrics are values associated with commodities bought or sold by the entity. Bought and sold commodities appear in separate records - one for the buyer and one for the seller. In addition to topology metrics, this table contains headroom properties, which are generated daily by headroom plans.

| Data Object  | Type                     | Nullable | Primary | Description                                                                                                         | Reference                   |
|--------------|--------------------------|----------|---------|---------------------------------------------------------------------------------------------------------------------|-----------------------------|
| time         | timestamp with time zone | False    | False   | When analysis discovered this metric. For headroom metrics, when the headroom plan was created.                     |                             |
| entity_oid   | bigint                   | False    | False   | The ID of the entity this metric pertains to.                                                                       |                             |
| type         | enum                     | False    | False   | The metric type.                                                                                                    | <a href="#">metric_type</a> |
| provider_oid | bigint                   | True     | False   | The ID of the entity that provides this metric. For example, for VMem, the ID of the host that provides the memory. |                             |

| Data Object   | Type             | Nullable | Primary | Description                                                     | Reference                   |
|---------------|------------------|----------|---------|-----------------------------------------------------------------|-----------------------------|
| key           | text             | True     | False   | The commodity key. This value can be null if no key is present. |                             |
| current       | double precision | True     | False   | Current utilization of commodity in provider entity.            |                             |
| capacity      | double precision | True     | False   | The capacity of the commodity in the provider entity.           |                             |
| utilization   | double precision | True     | False   | Overall utilization of the commodity in the provider entity.    |                             |
| consumed      | double precision | True     | False   | Amount of commodity currently used by consumer entity.          |                             |
| peak_current  | double precision | True     | False   | Peak amount of current used commodity in the provider entity.   |                             |
| peak_consumed | double precision | True     | False   | Peak amount of commodity currently used by the consumer entity. |                             |
| entity_type   | enum             | False    | False   | The type of the consumer entity.                                | <a href="#">entity_type</a> |

## Sample Use Cases

### VMem utilization for a Virtual Machine over time:

This query returns all the VM memory metrics collected for a specified virtual machine for a given time interval. The time interval is truncated to the hour.

```
SELECT m.time, avg(m.utilization) AS avg_util
FROM metric m,
 scope_to($__timeFrom()::timestampz - INTERVAL '24 hour', date_trunc('hour',
$__timeTo()::timestampz),
 'VIRTUAL_MACHINE', ARRAY[0]::bigint[], '0' = '0') s
WHERE m.entity_oid = s.oid
 and m.time BETWEEN s.from_time AND s.to_time
 AND m.type = 'VMEM'
 AND m.provider_oid IS NULL
 AND m.time BETWEEN $__timeFrom()::timestampz - interval '24 hour' AND date_trunc('hour',
$__timeTo()::timestampz)
GROUP BY 1
```

### Storage Amount of Physical Machines Running On a Cluster:

This query returns the max values of storage amounts for all the physical machines running in a cluster. Note that the only parameter given to the query other than the time range, is the `cluster_id`. We then have an internal function in Postgres that is able to translate the `cluster_id` to all the oids of the physical machines.

```
SELECT
date_trunc('day', m.time) as time,
m.entity_oid,
MAX(current) FILTER (WHERE m.type = 'STORAGE_AMOUNT') as stor_used,
MAX(capacity) FILTER (WHERE m.type = 'STORAGE_AMOUNT') as stor_cap,
MAX(current) FILTER (WHERE m.type = 'STORAGE_PROVISIONED') as stor_prov
FROM
metric m,
scope_to($__timeFrom(), $__timeTo(), 'STORAGE', ARRAY[$clusterId+0]::bigint[], $clusterId =
'0') s
WHERE
```

```

m.entity_oid = s.oid
AND m.time BETWEEN s.from_time AND s.to_time
AND m.type in ('STORAGE_AMOUNT', 'STORAGE_PROVISIONED')
AND m.time between $__timeFrom() and $__timeTo()
GROUP BY 1, 2

```

### Physical Machine Average Memory Utilization - Week Over Week By Day:

Returns the average daily memory utilization for virtual machines both for the current week and the past week.

```

WITH metrics AS (
 SELECT m.time, m.utilization
 FROM metric m,
 scope_to($__timeFrom()::timestampz - INTERVAL '7 day', $__timeTo(), 'PHYSICAL_MACHINE',
 ARRAY[0]::bigint[], '0' = '0') s
 WHERE
 m.entity_oid = s.oid
 AND m.time BETWEEN s.from_time AND s.to_time
 AND m.type = 'MEM'
 AND m.time BETWEEN ($__timeFrom()::TIMESTAMPZ - interval '7 day') AND date_trunc('day',
 $__timeTo()::TIMESTAMPZ)
 GROUP BY m.time, m.entity_oid, m.utilization
),
DATA AS (
 SELECT m.time, avg(m.utilization) AS avg_util
 FROM metrics m
 GROUP BY m.time
)
SELECT
 INTERVAL '1 DAY' + CASE
 WHEN time BETWEEN date_trunc('day', $__timeFrom()::TIMESTAMPZ - interval '7 day')
 AND date_trunc('day', $__timeTo()::TIMESTAMPZ - interval '7 day')
 THEN date_trunc('day', time + interval '7 day')
 ELSE date_trunc('day', time)
 END as "time",
 CASE
 WHEN time BETWEEN date_trunc('day', $__timeFrom()::TIMESTAMPZ - interval '7 day')
 AND date_trunc('day', $__timeTo()::TIMESTAMPZ - interval '7 day')
 THEN 'Previous Week'
 ELSE 'Week Ending ' || DATE($__timeTo()::TIMESTAMPZ)::TEXT
 END as metric,
 avg(avg_util)
FROM data
GROUP BY 1, 2
ORDER BY 1, 2

```

### Representation of Powered Off Virtual Machines:

Ideally, metrics of a turned off Virtual Machine will still be written on the table, assuming that the probe will send them in the entity dto. Most used values for commodities will have NULL values, we do this, instead of having 0's, in order not to affect potential averages over time, while the capacity should have the same value, independent of the state of the machine. Still, some used values for commodities won't be NULL, such as storage, since those are consumed by the machine even in a powered off state.

## pending\_action Table

This table lists the latest actions that have been recommended by the market. The table repopulates at a configurable interval (not necessarily every 10-minute market cycle). When it repopulates, all actions are replaced with the latest action recommendations.

This table does not include actions that are in progress.

| Data Object         | Type                     | Nullable | Primary | Description                            | Reference |
|---------------------|--------------------------|----------|---------|----------------------------------------|-----------|
| recommendation_time | timestamp with time zone | False    | False   | When analysis recommended this action. |           |

| Data Object       | Type             | Nullable | Primary | Description                                                                                                                             | Reference                       |
|-------------------|------------------|----------|---------|-----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| action_oid        | bigint           | False    | False   | The action ID. Use this to get more action details.                                                                                     |                                 |
| type              | enum             | False    | False   | the action type.                                                                                                                        | <a href="#">action_type</a>     |
| severity          | enum             | False    | False   | The action severity                                                                                                                     | <a href="#">severity</a>        |
| category          | enum             | False    | False   | The action category                                                                                                                     | <a href="#">action_category</a> |
| target_entity_id  | bigint           | False    | False   | The ID of the entity this action affects.                                                                                               |                                 |
| involved_entities | bigint[]         | True     | False   | An array of IDs for all the entities involved in the action. Includes the target entity.                                                |                                 |
| description       | text             | False    | False   | The text description of the action.                                                                                                     |                                 |
| savings           | double precision | False    | False   | The cost savings this action results in. A negative savings indicates an investment.                                                    |                                 |
| attrs             | jsonb            | True     | False   | Additional (often type-specific) info for the action. For example, in resizes this will contain the commodity type and from/to amounts. | <a href="#">action.attrs</a>    |

## Sample Use Cases

To get the target entities for all the pending actions, join this table with the entities table. This example lists the pending action type, entity name, and the entity type:

```
select pending_action.type AS action_type, entity.name, entity.type AS ent_type
FROM pending_action
INNER JOIN entity ON pending_action.target_entity_id=entity.oid
```

## pending\_action\_stats Table

| Data Object      | Type                     | Nullable | Primary | Description | Reference                        |
|------------------|--------------------------|----------|---------|-------------|----------------------------------|
| time             | timestamp with time zone | False    | False   |             |                                  |
| scope_oid        | bigint                   | False    | False   |             |                                  |
| entity_type      | enum                     | False    | False   |             | <a href="#">entity_type</a>      |
| environment_type | enum                     | False    | False   |             | <a href="#">environment_type</a> |

| Data Object           | Type | Nullable | Primary | Description | Reference |
|-----------------------|------|----------|---------|-------------|-----------|
| action_group          | int  | False    | False   |             |           |
| prior_action_count    | int  | False    | False   |             |           |
| cleared_action_count  | int  | False    | False   |             |           |
| new_action_count      | int  | False    | False   |             |           |
| involved_entity_count | int  | False    | False   |             |           |
| savings               | real | False    | False   |             |           |
| investments           | real | False    | False   |             |           |

## run\_health\_checks Table

This table lists the health checks that are current in the `{{ site.data.vars.Product_Short }}` notifications.

| Data Object | Type    | Nullable | Primary | Description                                       | Reference |
|-------------|---------|----------|---------|---------------------------------------------------|-----------|
| check_name  | text    | True     | False   | The display name of the health check.             |           |
| passed      | boolean | True     | False   | Whether the health check passed or failed.        |           |
| messages    | text[]  | True     | False   | The text description of the health check results. |           |

## scope Table

The Scope table tracks the association of two entities as they are related in each other's scope. The `seed_oid` identifies the containing entity (the entity that defines the scope), and the `scoped_oid` identifies the entity that is within the given scope. This table updates with each market cycle.

| Data Object | Type   | Nullable | Primary | Description                                                                                                                                                                                 | Reference |
|-------------|--------|----------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
| seed_oid    | bigint | False    | True    | The ID of the entity whose scope is the subject of the given record. For example, for a cluster-to-host relationship, the seed can be the cluster, and the scoped entity would be the host. |           |
| scoped_oid  | bigint | False    | True    | ID of the entity that is part of the seed scope.                                                                                                                                            |           |

| Data Object | Type                     | Nullable | Primary | Description                                                                                                                                                        | Reference                   |
|-------------|--------------------------|----------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------|
| scoped_type | enum                     | False    | False   | Entity type for the scoped entity.                                                                                                                                 | <a href="#">entity_type</a> |
| start       | timestamp with time zone | False    | True    | When this relationship first appeared in the topology. This covers the initial appearance, or an appearance after a gap (the relationship was temporarily broken). |                             |
| finish      | timestamp with time zone | False    | False   | WHEN this relationship LAST appeared IN THE TOPOLOGY. A value of 9999-12-31T23:59:59 means the relationship is not yet known to have terminated.                   |                             |

## Examples

### Example 1:

| seed_oid | scoped_oid | scoped_type      | start                | finish              |
|----------|------------|------------------|----------------------|---------------------|
| 111      | 999        | PHYSICAL_MACHINE | 2021-01-01T05:00:00Z | 9999-12-31T23:59:59 |
| 999      | 111        | COMPUTE_CLUSTER  | 2021-01-01T05:00:00Z | 9999-12-31T23:59:59 |

The first row indicates that cluster\_1 (oid: 111) contains Physical Machine\_1 (oid: 999). The start time indicates the first time the entity was discovered. This relationship is ongoing and this is represented by a finish timestamp of 9999-12-31T23:59:59. Row 2 shows that Physical Machine\_1 is contained by Cluster\_1.

### Example 2:

| seed_oid | scoped_oid | scoped_type      | start                | finish               |
|----------|------------|------------------|----------------------|----------------------|
| 111      | 999        | PHYSICAL_MACHINE | 2021-01-01T05:00:00Z | 2021-01-31T05:00:00Z |
| 999      | 111        | COMPUTE_CLUSTER  | 2021-01-01T05:00:00Z | 2021-01-31T05:00:00Z |

| seed_oid | scoped_oid | scoped_type      | start                | finish              |
|----------|------------|------------------|----------------------|---------------------|
| 222      | 999        | PHYSICAL_MACHINE | 2021-01-31T05:00:00Z | 9999-12-31T23:59:59 |
| 999      | 222        | COMPUTE_CLUSTER  | 2021-01-31T05:00:00Z | 9999-12-31T23:59:59 |

These records show that Physical\_Machine\_1 moved to Cluster\_2 (oid: 2222) on January 31. Note that the timestamp for the end of the relationship with Cluster\_1 is the same as the timestamp for the start of the relationship with Cluster\_2.

## Sample Use Cases

### Count Physical Machines in Clusters Over A Given Time Interval

```

WITH all_relationships AS (
 select
 e.name cluster_name, s.seed_oid cluster_oid, s.scoped_oid pm_oid
 FROM
 entity e
 JOIN scope s on e.oid = s.seed_oid
 and s.scoped_type = 'PHYSICAL_MACHINE'
 WHERE e.type = 'COMPUTE_CLUSTER'
 AND ('2021-02-01T05:00:00Z', '2021-02-18T04:59:59Z') OVERLAPS (s.start, s.finish)
), non_duplicate_relationships AS (
 -- The same relationships between 2 entities can exist in given time interval if they move
 -- in and out of each others scopes
 -- We account for these duplicates entries at varying time intervals
 select *
 from all_relationships
 GROUP BY cluster_name, cluster_oid, pm_oid
)
select cluster_name, count(*)
from non_duplicate_relationships
group by cluster_name

```

## scope\_to Table

| Data Object | Type                     | Nullable | Primary | Description | Reference |
|-------------|--------------------------|----------|---------|-------------|-----------|
| oid         | bigint                   | True     | False   |             |           |
| from_time   | timestamp with time zone | True     | False   |             |           |
| to_time     | timestamp with time zone | True     | False   |             |           |
| seed        | bigint                   | True     | False   |             |           |

## scope\_to\_types Table

| Data Object | Type                     | Nullable | Primary | Description | Reference |
|-------------|--------------------------|----------|---------|-------------|-----------|
| oid         | bigint                   | True     | False   |             |           |
| from_time   | timestamp with time zone | True     | False   |             |           |
| to_time     | timestamp with time zone | True     | False   |             |           |
| seed        | bigint                   | True     | False   |             |           |



| Data Object | Type | Nullable | Primary | Description | Reference                   |
|-------------|------|----------|---------|-------------|-----------------------------|
| e_type      | enum | True     | False   |             | <a href="#">entity_type</a> |

## topology\_stats Table

| Data Object | Type                     | Nullable | Primary | Description | Reference |
|-------------|--------------------------|----------|---------|-------------|-----------|
| time        | timestamp with time zone | False    | False   |             |           |
| attrs       | jsonb                    | True     | False   |             |           |

## wasted\_file Table

This table stores the paths to the current wasted files in the environment, as well as properties of the files such as last modification or file size.

| Data Object       | Type                     | Nullable | Primary | Description                                           | Reference |
|-------------------|--------------------------|----------|---------|-------------------------------------------------------|-----------|
| path              | text                     | True     | False   | The file path.                                        |           |
| file_size_kb      | bigint                   | True     | False   | The file size.                                        |           |
| modification_time | timestamp with time zone | True     | False   | When the file was last modified.                      |           |
| storage_oid       | bigint                   | True     | False   | The ID of the storage that contains the file.         |           |
| storage_name      | text                     | True     | False   | The name of the storage (denormalized for reporting). |           |

## Sample Use Cases

### Get information about storage files for a data store:

The query returns the storage name, the file path, the site of the file and the last modified time for each file of a data store. The query accepts the \$storage\_oid as a parameter.

```
SELECT storage_name as "Datastore",
 path as "File Path",
 file_size_kb as "File Size",
 modification_time as "Last Modified Time"
FROM wasted_file
WHERE file_size_kb >= 0
 AND ARRAY[storage_oid]::text[] && Array['$storage_oid']::text[]
ORDER BY 3 DESC
```

## JSON Data Objects

The following JSON objects represent data about specific entities in your environment.

### ActionCommodity JSON

The capacity and utilization of a commodity that is affected by a given action.

| Data Object           | Description                                                                           | Type   |
|-----------------------|---------------------------------------------------------------------------------------|--------|
| capacity              | The available capacity of the given commodity.                                        | float  |
| percentileUtilization | For entities and resources that support it, the calculated percentile of utilization. | Double |
| used                  | The amount of the resource currently in use.                                          | float  |
| utilization           | The percentage of capacity in use.                                                    | float  |

## ActionEntity JSON

For a given action, a description of the entity this action changes.

| Data Object | Description                                                                        | Type                        | Reference                   |
|-------------|------------------------------------------------------------------------------------|-----------------------------|-----------------------------|
| oid         | The ID of the entity.                                                              | Long                        |                             |
| name        | The entity name. This corresponds to the name that displays in the user interface. | String                      |                             |
| type        | The entity type.                                                                   | <a href="#">entity_type</a> | <a href="#">entity_type</a> |

## ActionImpactedCost JSON

For cloud entities, the on-demand costs or RI coverage that is affected by a given action. The action shows one object for the cost before the action, and another for the cost after the action.

| Data Object          | Description                              | Type  |
|----------------------|------------------------------------------|-------|
| onDemandCost         | For on-demand billing, the entity cost.  | Float |
| onDemandRate         | For on-demand billing, the rate of cost. | Float |
| riCoveragePercentage | For RIs, the percentage of RI coverage.  | Float |

## ActionImpactedCosts JSON

For a given action, descriptions of the Before and After costs.

| Data Object   | Description                                                       | Type                               | Reference                          |
|---------------|-------------------------------------------------------------------|------------------------------------|------------------------------------|
| afterActions  | A description of what the cost will be after the action executes. | <a href="#">ActionImpactedCost</a> | <a href="#">ActionImpactedCost</a> |
| beforeActions | A description of the current cost, before the action executes.    | <a href="#">ActionImpactedCost</a> | <a href="#">ActionImpactedCost</a> |

## ActionImpactedEntity JSON

For a given action, a description of an entity that is affected by that action.

| Data Object     | Description                                                                                                   | Type                                | MapKeyType                  | Reference                           |
|-----------------|---------------------------------------------------------------------------------------------------------------|-------------------------------------|-----------------------------|-------------------------------------|
| affectedCosts   | A description of the before and after costs for this entity.                                                  | <a href="#">ActionImpactedCosts</a> |                             | <a href="#">ActionImpactedCosts</a> |
| affectedMetrics | A description of the before and after metrics for this entity.                                                | <a href="#">ImpactedMetric</a>      | <a href="#">metric_type</a> | <a href="#">ImpactedMetric</a>      |
| name            | The entity name.                                                                                              | String                              |                             |                                     |
| oid             | The entity ID.                                                                                                | Long                                |                             |                                     |
| settings        | For entities that support percentile calculations, the aggressiveness and observation period for this entity. | <a href="#">EntitySettings</a>      |                             | <a href="#">EntitySettings</a>      |
| type            | The entity type.                                                                                              | <a href="#">entity_type</a>         |                             | <a href="#">entity_type</a>         |

## BuyRIInfo JSON

The information that analysis can use to calculate Buy RI actions.

| Data Object   | Description                                                                                                              | Type                         | Reference                    |
|---------------|--------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------|
| computeTier   | The compute tier, or workload family for the action. The ActionEntity object will be for an entity of type COMPUTE_TIER. | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| count         |                                                                                                                          | Integer                      |                              |
| masterAccount | The account that manages the RI billing.                                                                                 | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| region        | The region that will host the RIs.                                                                                       | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |
| target        | The cloud account or subscription that will manage the RIs.                                                              | <a href="#">ActionEntity</a> | <a href="#">ActionEntity</a> |

## CommodityChange JSON

The change (from - to) of an entity's commodity that is proposed for a given action.

| Data Object   | Description                                                             | Type                        | Reference                   |
|---------------|-------------------------------------------------------------------------|-----------------------------|-----------------------------|
| attribute     |                                                                         | String                      |                             |
| commodityType | The type of commodity the action will change - MEM, IO_THROUGHPUT, etc. | <a href="#">metric_type</a> | <a href="#">metric_type</a> |

| Data Object      | Description                                                                                                                             | Type                                      | Reference                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|-------------------------------------------|
| from             | The commodity value before the action executes.                                                                                         | Float                                     |                                           |
| percentileChange | For entities and commodities that support percentile calculations, the before and after values for percentile utilization and settings. | <a href="#">CommodityPercentileChange</a> | <a href="#">CommodityPercentileChange</a> |
| target           | A description of the entity this action will change.                                                                                    | <a href="#">ActionEntity</a>              | <a href="#">ActionEntity</a>              |
| to               | The commodity value after the action executes.                                                                                          | Float                                     |                                           |
| unit             | The displayed units of measurement.                                                                                                     | String                                    |                                           |

## CommodityPercentileChange JSON

If an action changes a commodity that participates in percentile-based analysis, the percentile settings, and the commodity values before and after the action.

| Data Object           | Description                                                  | Type    |
|-----------------------|--------------------------------------------------------------|---------|
| after                 | The percentile utilization after the action executes         | Double  |
| aggressiveness        | The aggressiveness setting used to calculate the percentile. | Integer |
| before                | The percentile utilization before the action executes.       | Double  |
| observationPeriodDays | The observation period used to calculate the percentile.     | Integer |

## DeleteInfo JSON

A description of a file is unattached and should be deleted.

| Data Object           | Description                                 | Type    | Format                   |
|-----------------------|---------------------------------------------|---------|--------------------------|
| filePath              | The path to the file.                       | String  |                          |
| fileSize              | The size of the file.                       | Double  |                          |
| lastModifiedTimestamp | When the file was last modified.            | String  | yyyy-MM-dd'T'HH:mm:ss'Z' |
| unattachedDays        | How many days the file has been unattached. | Integer |                          |

| Data Object | Description                                      | Type   | Format |
|-------------|--------------------------------------------------|--------|--------|
| unit        | The displayed unit of measure for the file size. | String |        |

## entity.attrs JSON

Additional descriptions for a given entity. The values an instance of this object can have are different for different entity types.

| Data Object        | Description                                              | Type                   | SupportedEntityTypes | Repeated | Reference              |
|--------------------|----------------------------------------------------------|------------------------|----------------------|----------|------------------------|
| attachment_state   | Whether a given volume is attached or unattached.        | <u>AttachmentState</u> | VIRTUAL_VOLUME       | False    | <u>AttachmentState</u> |
| connected_networks | A list of network names that a given VM is connected to. | MULTI_TEXT             | VIRTUAL_MACHINE      | True     |                        |
| cpu_model          | For a physical host, the model of CPU on that host.      | TEXT                   | PHYSICAL_MACHINE     | False    |                        |
| guest_os_name      | The name of the OS running on a given VM.                | TEXT                   | VIRTUAL_MACHINE      | False    |                        |
| guest_os_type      | The specific type for the OS running on a given VM.      | <u>OSType</u>          | VIRTUAL_MACHINE      | False    | <u>OSType</u>          |
| is_local           | Whether a storage is local to the machine, or networked. | BOOLEAN                | STORAGE              | False    |                        |
| model              | The specific model of the given physical host.           | TEXT                   | PHYSICAL_MACHINE     | False    |                        |
| num_cpus           | The number of CPUs on the given physical host.           | NUMBER                 | PHYSICAL_MACHINE     | False    |                        |

| Data Object | Description                                                     | Type                                                           | SupportedEntityTypes               | Repeated  | Reference              |
|-------------|-----------------------------------------------------------------|----------------------------------------------------------------|------------------------------------|-----------|------------------------|
| tags        | A list of tags that are assigned to the given entity.           | TEXT_MULTIMEDIA P (Embedded Reporting) or List (Data Exporter) | ALL                                | False     |                        |
| targets     | The target that manages the given entity.                       | <a href="#">Target</a>                                         | ALL                                | True      | <a href="#">Target</a> |
| timezone    | The timezone that is set for the clock on the physical machine. | TEXT                                                           | PHYSICAL_MACHINE                   | False     |                        |
| vendor_id   |                                                                 | TEXT                                                           | VIRTUAL_VOLUME<br>BUSINESS_ACCOUNT | Reference |                        |

## EntitySettings JSON

For an entity that participates in percentile-based analysis, the aggressiveness setting and observation period.

| Data Object                     | Description                         | Type    |
|---------------------------------|-------------------------------------|---------|
| percentileAggressiveness        | The current aggressiveness setting. | Integer |
| percentileObservationPeriodDays | The current observation period.     | Integer |

## group.attrs JSON

Additional descriptions for a given group.

| Data Object | Description                                                | Type    | Repeated | Reference | SupportedEntityTypes                                                                           | SupportedGroupTypes                                                                            |
|-------------|------------------------------------------------------------|---------|----------|-----------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| dynamic     | Whether this group is static or dynamic. true for dynamic. | BOOLEAN | False    |           | STORAGE_CLUSTER<br>COMPUTE_CLUSTER<br>GROUP<br>K8S_CLUSTER<br>BILLING_FAMILY<br>RESOURCE_GROUP | RESOURCE_GROUP<br>COMPUTE_CLUSTER<br>BILLING_FAMILY<br>GROUP<br>K8S_CLUSTER<br>STORAGE_CLUSTER |

| Data Object  | Description | Type                                                             | Repeated | Reference | SupportedEntityTypes                                                                           | SupportedGroupTypes                                                                            |
|--------------|-------------|------------------------------------------------------------------|----------|-----------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| member_types |             | MULTI_TEXT                                                       | True     |           | STORAGE_CLUSTER<br>COMPUTE_CLUSTER<br>GROUP<br>K8S_CLUSTER<br>BILLING_FAMILY<br>RESOURCE_GROUP | RESOURCE_GROUP<br>COMPUTE_CLUSTER<br>BILLING_FAMILY<br>GROUP<br>K8S_CLUSTER<br>STORAGE_CLUSTER |
| origin       |             | TEXT                                                             | False    |           | STORAGE_CLUSTER<br>COMPUTE_CLUSTER<br>GROUP<br>K8S_CLUSTER<br>BILLING_FAMILY<br>RESOURCE_GROUP | RESOURCE_GROUP<br>COMPUTE_CLUSTER<br>BILLING_FAMILY<br>GROUP<br>K8S_CLUSTER<br>STORAGE_CLUSTER |
| tags         |             | TEXT_MULTIMAP<br>(Embedded Reporting) or<br>List (Data Exporter) | False    |           | GROUP<br>RESOURCE_GROUP<br>COMPUTE_CLUSTER<br>K8S_CLUSTER<br>STORAGE_CLUSTER<br>BILLING_FAMILY | ALL                                                                                            |

| Data Object | Description | Type                   | Repeated | Reference              | SupportedEntityTypes                                                                           | SupportedGroupTypes |
|-------------|-------------|------------------------|----------|------------------------|------------------------------------------------------------------------------------------------|---------------------|
| targets     |             | <a href="#">Target</a> | False    | <a href="#">Target</a> | GROUP<br>RESOURCE_GROUP<br>COMPUTE_CLUSTER<br>K8S_CLUSTER<br>STORAGE_CLUSTER<br>BILLING_FAMILY | ALL                 |

## ImpactedMetric JSON

For a metric on an entity that is affected by an action, a description of the impacted metric.

| Data Object   | Description                                                                     | Type                            | Reference                       |
|---------------|---------------------------------------------------------------------------------|---------------------------------|---------------------------------|
| afterActions  | A description of what the metric utilization will be after the action executes. | <a href="#">ActionCommodity</a> | <a href="#">ActionCommodity</a> |
| beforeActions | A description of the current utilization, before the action executes.           | <a href="#">ActionCommodity</a> | <a href="#">ActionCommodity</a> |

## MoveChange JSON

For a vMotion action, the changed placement of the affected entity.

| Data Object | Description                      | Type                                 | Repeated | Reference                            |
|-------------|----------------------------------|--------------------------------------|----------|--------------------------------------|
| from        | Where the entity will move from. | <a href="#">ActionImpactedEntity</a> | False    | <a href="#">ActionImpactedEntity</a> |
| resource    |                                  | <a href="#">ActionEntity</a>         | True     | <a href="#">ActionEntity</a>         |
| to          | Where the entity will move to.   | <a href="#">ActionImpactedEntity</a> | False    | <a href="#">ActionImpactedEntity</a> |

## Target JSON

| Data Object    | Description | Type   |
|----------------|-------------|--------|
| category       |             | String |
| entityVendorId |             | String |
| name           |             | String |
| oid            |             | Long   |
| type           |             | String |



## action.attrs JSON

Extra data that pertains to a given action.

| JSON Object | Description                                                               | Type                                 | MapKeyType                  | Reference                            |
|-------------|---------------------------------------------------------------------------|--------------------------------------|-----------------------------|--------------------------------------|
| buyRiInfo   | For an action to buy RIs, additional information about the RIs.           | <a href="#">BuyRiInfo</a>            |                             | <a href="#">BuyRiInfo</a>            |
| deleteInfo  | For an action to delete a file, a description of that file.               | <a href="#">DeleteInfo</a>           |                             | <a href="#">DeleteInfo</a>           |
| moveInfo    | For a move action, a description of from and to for the move.             | <a href="#">MoveChange</a>           | <a href="#">entity_type</a> | <a href="#">MoveChange</a>           |
| resizeInfo  | For a resize action, a description of the commodity changes.              | <a href="#">CommodityChange</a>      | <a href="#">metric_type</a> | <a href="#">CommodityChange</a>      |
| scaleInfo   | For scaling a cloud VM, a description of the from and to for the VM move. | <a href="#">MoveChange</a>           | <a href="#">entity_type</a> | <a href="#">MoveChange</a>           |
| target      | Extra information about the entity this action impacts.                   | <a href="#">ActionImpactedEntity</a> |                             | <a href="#">ActionImpactedEntity</a> |

## Enums

---

These enums specify values for certain states or types.

### AttachmentState Enum

Whether a given is attached or unattached.

- ATTACHED:
- UNATTACHED:

### OSType Enum

The OS types that Turbonomic discovers on hosts and VMs.

- LINUX:
- LINUX\_WITH\_SQL\_ENTERPRISE:
- LINUX\_WITH\_SQL\_STANDARD:
- LINUX\_WITH\_SQL\_WEB:
- RHEL:
- SUSE:
- UBUNTU\_PRO:
- UNKNOWN\_OS:
- WINDOWS:
- WINDOWS\_BYOL:

- WINDOWS\_WITH\_SQL\_ENTERPRISE:
- WINDOWS\_WITH\_SQL\_STANDARD:
- WINDOWS\_WITH\_SQL\_WEB:

## **action\_category Enum**

The different categories for a given action.

- COMPLIANCE:
- EFFICIENCY\_IMPROVEMENT:
- PERFORMANCE\_ASSURANCE:
- PREVENTION:
- UNKNOWN:

## **action\_mode Enum**

The different modes for an action, such as automated or manual.

- AUTOMATIC:
- DISABLED:
- EXTERNAL\_APPROVAL:
- MANUAL:
- RECOMMEND:

## **action\_state Enum**

The different states for an action, such as completed or pending.

- ACCEPTED:
- CLEARED:
- FAILED:
- FAILING:
- IN\_PROGRESS:
- POST\_IN\_PROGRESS:
- PRE\_IN\_PROGRESS:
- QUEUED:
- READY:
- REJECTED:
- SUCCEEDED:

## **action\_type Enum**

The different types of action.

- ACTIVATE:
- ALLOCATE:
- BUY\_RI:
- DEACTIVATE:
- DELETE:
- MOVE:
- NONE:

- PROVISION:
- RECONFIGURE:
- RESIZE:
- SCALE:
- START:
- SUSPEND:

## **attr\_type Enum**

The supported types of attributes. For each type, you can find different data stored in the attribute.

- ENTITY\_STATE:
- NUM\_VCPU:
- VOLUME\_ATTACHED:

## **cost\_category Enum**

For different cost data, the service or category that the cost pertains to.

- IP:
- ON\_DEMAND\_COMPUTE:
- ON\_DEMAND\_LICENSE:
- RESERVED\_LICENSE:
- RI\_COMPUTE:
- SPOT:
- STORAGE:
- TOTAL:

## **cost\_source Enum**

The type of billing agreement that is the source of the given cost data.

- BUY\_RI\_DISCOUNT:
- ENTITY\_UPTIME\_DISCOUNT:
- ON\_DEMAND\_RATE:
- RI\_INVENTORY\_DISCOUNT:
- TOTAL:
- UNCLASSIFIED:

## **entity\_state Enum**

The running state for an entity, such as powered off, suspended, or running.

- FAILOVER:
- MAINTENANCE:
- POWERED\_OFF:
- POWERED\_ON:
- SUSPENDED:
- UNKNOWN:

## entity\_type Enum

The set of supported entities that Turbonomic manages.

- ACTION\_MANAGER:
- APPLICATION:
- APPLICATION\_COMPONENT:
- APPLICATION\_SERVER:
- AVAILABILITY\_ZONE:
- BILLING\_FAMILY:
- BUSINESS:
- BUSINESS\_ACCOUNT:
- BUSINESS\_APPLICATION:
- BUSINESS\_ENTITY:
- BUSINESS\_TRANSACTION:
- BUSINESS\_USER:
- CHASSIS:
- CLOUD\_COMMITMENT:
- CLOUD\_SERVICE:
- COMPUTE\_CLUSTER:
- COMPUTE\_RESOURCE:
- COMPUTE\_TIER:
- CONTAINER:
- CONTAINER\_PLATFORM\_CLUSTER:
- CONTAINER\_POD:
- CONTAINER\_SPEC:
- DATABASE:
- DATABASE\_SERVER:
- DATABASE\_SERVER\_TIER:
- DATABASE\_TIER:
- DATACENTER:
- DESIRED\_RESERVED\_INSTANCE:
- DESKTOP\_POOL:
- DISK\_ARRAY:
- DISTRIBUTED\_VIRTUAL\_PORTGROUP:
- DPOD:
- GROUP:
- HCI\_PHYSICAL\_MACHINE:
- HYPERVISOR\_SERVER:
- INFRASTRUCTURE:
- INTERNET:
- IO\_MODULE:
- IP:
- K8S\_CLUSTER:
- LICENSING\_SERVICE:

- LOAD\_BALANCER:
- LOGICAL\_POOL:
- MAC:
- MOVER:
- NAMESPACE:
- NETWORK:
- NETWORKING\_ENDPOINT:
- NODE\_POOL:
- OPERATOR:
- PHYSICAL\_MACHINE:
- PORT:
- PROCESSOR\_POOL:
- REGION:
- RESERVED\_INSTANCE:
- RESERVED\_INSTANCE\_SPECIFICATION:
- RESOURCE\_GROUP:
- RIGHT\_SIZER:
- SAVINGS:
- SERVICE:
- SERVICE\_ENTITY\_TEMPLATE:
- SERVICE\_PROVIDER:
- STORAGE:
- STORAGE\_CLUSTER:
- STORAGE\_CONTROLLER:
- STORAGE\_TIER:
- STORAGE\_VOLUME:
- SWITCH:
- TARGET:
- THIS\_ENTITY:
- THREE\_TIER\_APPLICATION:
- VIEW\_POD:
- VIRTUAL\_APPLICATION:
- VIRTUAL\_DATACENTER:
- VIRTUAL\_MACHINE:
- VIRTUAL\_VOLUME:
- VLAN:
- VM\_SPEC:
- VPOD:
- WEB\_SERVER:
- WORKLOAD\_CONTROLLER:
- \_NONE\_:

## environment\_type Enum

Whether the environment is cloud, on-prem, or hybrid.

- CLOUD:
- HYBRID:
- ON\_PREM:
- UNKNOWN\_ENV:

## file\_type Enum

Different types of file, such as configuration file or ISO image.

- CONFIGURATION:
- DISK:
- ESXCONSOLE:
- ISO:
- LOG:
- MEMORY:
- SNAPSHOT:
- SWAP:

## metric\_type Enum

The different types of metrics that Turbonomic discovers for entities.

- ACTIVE\_SESSIONS:  
Count of the number of actively logged in sessions  
Units: Count
- BALLOONING:  
Number of kb the host is dedicated for ballooning  
Units: kilobytes
- BUFFER\_COMMODITY:  
Utilization of a port buffer (fabric, UCS specific)  
Units: megabytes
- CONNECTION:  
Number of connections onto the application  
Units: Count
- COOLING:
- CPU:  
For Physical machine - usage of CPU (current column). For VM - consumed CPU on the host (consumed column).  
Units: MHz
- CPU\_ALLOCATION:  
For VMs (consumed column) - VM buys from VDC or from desktop pool (VDI). For PMs - PMs sells to VDCs.  
Units: MHz
- CPU\_HEADROOM:

Number of VMs that can fit in all of the available CPU (on a cluster)

Units: Count

- CPU\_PROVISIONED:

Capacity is based on the over-provisioned value in policy

Units: MHz

- CPU\_READY:

Ready queue

Units: ms

- DB\_CACHE\_HIT\_RATE:

The percentage of accesses that result in cache hits.

Units: Percentage

- DB\_MEM:

DB mem utilization or DM capacity

Units: kilobytes

- DTU:

Capacity defined in the cloud

Units: Count

- EXTENT:

Internal

Units: NA

- FLOW:

Amount of network traffic per second between two nodes

Units: kilobytes/sec

- FLOW\_ALLOCATION:

Units: bytes

- HEAP:

Amount of heap configured for an application

Units: kilobytes

- IMAGE\_CPU:

For VDI, usage of the business user

Units: MHz

- IMAGE\_MEM:

For VDI, usage of the business user

Units: kilobytes

- IMAGE\_STORAGE:

For VDI, usage of the business user

Units: megabytes

- IO\_THROUGHPUT:

Amount of traffic through the IO adapter

Units: kilobytes/sec

- MEM:  
Physical machine - usage of memory. VM - consumed memory.  
Units: kilobytes
- MEM\_ALLOCATION:  
Physical machine - what the host provides to VDCs. VM (consumed column) - what the VM buys from VDC or desktop pool.  
Units: kilobytes
- MEM\_HEADROOM:  
Number of VMs that can fit in the available memory on the cluster  
Units: Count
- MEM\_PROVISIONED:  
Capacity, based on the over-provisioned value in policy  
Units: kilobytes
- NET\_THROUGHPUT:  
Amount of traffic through the Network adapter  
Units: kilobytes/sec
- POOL\_CPU:  
For VDI, usage of the desktop pool  
Units: MHz
- POOL\_MEM:  
For VDI, usage of the desktop pool  
Units: kilobytes
- POOL\_STORAGE:  
For VDI, usage of the desktop pool  
Units: megabytes
- PORT\_CHANNEL:  
Traffic on ports for UCS Mgr/HPE1  
Units: kilobytes/sec
- POWER:
- Q16\_VCPU:  
VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.  
Units: ms
- Q1\_VCPU:  
VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.  
Units: ms
- Q2\_VCPU:



VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q32\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q3\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q4\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q5\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q64\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q6\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q7\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- Q8\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- QN\_VCPU:

VM - Sum of the Ready Queue usage of all of the CPU on the VM. To normalize the value, divide it by the number of vCPU. Host - Average of the underlying Ready Queue values from all of the corresponding VMs with the same number of vCPUs.

Units: ms

- REMAINING\_GC\_CAPACITY:

Percentage of time not spent on garbage collection

Units: Percentage

- RESPONSE\_TIME:

Application response

Units: ms

- SLA\_COMMODITY:

User-defined value

Units: Number

- STORAGE\_ACCESS:

Count of the number of I/O (read + write) within two discovery cycles

Units: count

- STORAGE\_ALLOCATION:

Allocated storage

Units: megabytes

- STORAGE\_AMOUNT:

Used storage

Units: megabytes

- STORAGE\_HEADROOM:

Count of VM that can fit for the storage in a cluster in terms of storage

Units: Count

- STORAGE\_LATENCY:

Acceptable latency. User can set capacity for most storage, is discovered for NetApp.

Units: ms

- STORAGE\_PROVISIONED:

Capacity, based on the over-provisioned value in policy

Units: megabytes

- SWAPPING:

Memory swapping

Units: bytes/sec

- THREADS:

Threads in use

Units: Count

- TOTAL\_HEADROOM:

Min value of CPU, Mem, IO, of the headrooms

Units: Count

- TOTAL\_SESSIONS:

- Total number of possible sessions
- Units: Count
- TRANSACTION:
  - Transactions for the application
  - Units: transactions/sec
- TRANSACTION\_LOG:
  - Size of the transaction log
  - Units: megabytes
- VCPU:
  - CPU for the VM
  - Units: MHz
- VCPU\_LIMIT\_QUOTA:
  - Units: millicores
- VCPU\_REQUEST:
  - Units: millicores
- VCPU\_REQUEST\_QUOTA:
  - Units: millicores
- VMEM:
  - Memory sold by a VM. Memory sold and bought for a container
  - Units: kilobytes
- VMEM\_LIMIT\_QUOTA:
  - Units: kilobytes
- VMEM\_REQUEST:
  - Units: kilobytes
- VMEM\_REQUEST\_QUOTA:
  - Units: kilobytes
- VSTORAGE:
  - Units: megabytes

## savings\_type Enum

The different types of savings that Turbonomic actions can give.

- MISSED\_INVESTMENTS:
  - Actions you could have taken that would increase your investment to provide more resources to your applications.
- MISSED\_SAVINGS:
  - Actions you could have taken that could have reduced your costs.
- REALIZED\_INVESTMENTS:
  - Actions you have taken that increase your investment to provide more resources to your applications.
- REALIZED\_SAVINGS:
  - Actions you have taken that have reduced your costs.

## severity Enum

For a given action, the severity of the issue that it addresses.

- CRITICAL:
- MAJOR:
- MINOR:
- NORMAL:

## terminal\_state Enum

The ultimate state of an action.

- FAILED:
- SUCCEEDED:

## Configuring an External TimescaleDB

---

You can configure External Reports to use an external deployment of TimescaleDB. To enable an external deployment, you will:

- Deploy or access the external TimescaleDB instance that you will use This is a PostgreSQL server with the TimescaleDB extension. The DB server must be deployed in a way that is accessible to your installation of Turbonomic. It can be deployed on the cloud or in an on-prem VM. The Embedded Reports feature currently supports PostgreSQL 12.x and TimescaleDB 2.0.1. For more information, see:
  - [Creating an External TimescaleDB](#)
  - See the article, <https://docs.timescale.com/v2.0/getting-started/installation>
- Provision users, databases, and schemas on the TimescaleDB instance To provision these database objects you can manually provision or enable Turbonomic to automatically provision. For more information, see:
  - [Manual Provisioning of Users and Databases](#)
  - [Automatic Provisioning of Users and Databases](#)
- Edit properties in the Turbonomic cr.yaml file. Note that the location of the cr.yaml file is different, depending on whether you are configuring an OVA installation or a Kubernetes Node installation of Turbonomic.
- Optionally, enable secure connections between Turbonomic and the Timescale DB instance For information about enabling TLS/SSL for database connections, see the PostgreSQL documentation [here](#).

We have verified that the steps to create a self-signed certificate, as outlined in that document, produce a server that functions properly with all components of Embedded Reporting. However, we do not recommend using self-signed certificates in production instances.

To configure External Reports to use an external deployment of TimescaleDB, you must edit the Turbonomic cr.yaml file. .

## Creating an External TimescaleDB

You can create an external TimescaleDB service that you host in your environment, or you can use an existing TimescaleDB service that has been deployed on the cloud or in your enterprise by the DB Management team.

The TimescaleDB service that Turbonomic uses must satisfy these conditions:

- DeploymentTurbonomicThe TimescaleDB is a PostgreSQL server with the TimescaleDB extension. It must be deployed in a way that is accessible to your installation of Turbonomic. It can be deployed on the cloud or in an on-prem VM.

For information about installing Timescale, see the article: <https://docs.timescale.com/v2.0/getting-started/installation>.

- Supported VersionTurbonomicThe Embedded Reports feature currently supports PostgreSQL 12.x and TimescaleDB 2.0.1.
- An account with global R/W privileges on the TimescaleDB instanceTurbonomicThe Embedded Reports feature requires a specific set of databases to be created on the TimescaleDB. It is typical practice for an administrator to manually create the necessary databases. Alternatively, you can create a global R/W user account for the Turbonomic platform so it can create the databases automatically.
- Entry PointTurbonomicThe TimescaleDB must provide an entry point via DNS Name or IP Address that you can access from Turbonomic. You will configure this access in the Turbonomic cr.yaml file.

To create the TimescaleDB:

1. Install the DB package.TurbonomicFor information about installing Timescale, see the articles: <https://docs.timescale.com/v2.0/getting-started/installation>.
2. Create a global R/W user account that can be used to create databases on the DB service.TurbonomicLaunch the DB service and open a command session. Then create the global R/W account.TurbonomicYou can alternatively set credentials to this account in the Turbonomic cr.yaml file. You can enter them as cleartext, or you can manage the credentials via Kubernetes Secrets. If you set the account credentials in cr.yaml, {{ site.data.vars.Product\_Short }} can create the databases that the Embedded Reports feature needs.
3. (OPTIONAL) Manually create the users and databases that Embedded Reports will use.TurbonomicIf you cannot grant Turbonomic global R/W access to the DB, you must manually create the databases and users for Embedded Reports.

For information about manually creating databases and users, see [Manual Provisioning of Users and Databases](#).

For information about automatically creating databases and users via R/W access, see [Automatic Provisioning of Users and Databases](#).

## Manual Provisioning of Users and Databases

woof

For Turbonomic to automatically create the required databases for Embedded Reports, the Turbonomic cr.yaml file must specify credentials for a user account that has global R/W access to the DB service. To do this you can Store the credentials in cleartext in the cr.yaml file.

If Automatic Provisioning is appropriate for your environment, then you can skip this section and go directly to [Automatic Provisioning of Users and Databases](#). That is where you will specify the global R/W account that Embedded Reports can use to access your TimescaleDB.

*If Automatic Provisioning is not appropriate for your environment*, then you must access the DB service with a R/W account and manually add the databases and user accounts that Embedded Reports can use. Then, after you have added the databases to your TimescaleDB, you will edit the Turbonomic cr.yaml file to make use of these databases.

### Manually Provisioning Databases and Users on the TimescaleDB

The following steps will properly provision the database objects that Embedded Reporting requires. Note that these examples use default names for illustration. You can substitute your own names for databases and users. You must remember the names you use, because you must match that naming as you edit the Turbonomic cr.yaml file.

To provision the databases and users, open a command session on the TimescaleDB, and execute the following commands:

- Create two databases - one for extractor data, one for grafana data: The extractor database manages the Turbonomic data stream, and the grafana database manages data for reporting within Grafana.

Execute the following commands, where you can provide your own database names instead of `extractor` and `grafana`:

```
CREATE DATABASE extractor;
CREATE DATABASE grafana;
```

- Create database users: For the extractor database, you will create a R/W user, a read-only group, and read-only user. You will also create a R/W user for the grafana database.

Execute the following commands, where you can provide your own names for instances of USER or ROLE:

```
-/- main read/write user for extractor data
CREATE USER extractor PASSWORD '<password>';
-/- group for users with read-only access to extractor data
CREATE ROLE readers_extractor_extractor;
-/- read-only user for extractor data as a member of that group
CREATE USER query PASSWORD '<password>';
GRANT CONNECT ON DATABASE extractor TO readers_extractor_extractor;
GRANT readers_extractor_extractor TO query;
-/- read-write user for grafana data
CREATE USER grafana_backend PASSWORD '<password>';
```

- Create and prepare the schema for extractor data Connect to the extractor database and execute these commands, where you can provide your own name for the SCHEMA, and you grant privileges to the users and roles you created above:

```
CREATE SCHEMA extractor;
-/- read/write user has full access
GRANT ALL PRIVILEGES ON SCHEMA extractor TO extractor;
-/- all users in readers group have read-only access
GRANT USAGE on SCHEMA extractor TO readers_extractor_extractor;
GRANT SELECT ON ALL TABLES IN SCHEMA extractor TO readers_extractor_extractor;
-/- make sure readers get access to any tables added in the future
ALTER DEFAULT PRIVILEGES IN SCHEMA extractor GRANT SELECT ON TABLES TO
readers_extractor_extractor;
-/- make the extractor and query users use the extractor schema by default
ALTER ROLE extractor SET search_path TO 'extractor';
ALTER ROLE query SET search_path TO 'extractor';
-/- install the timescaledb plugin into the extractor database using the extractor schema
CREATE EXTENSION timescaledb SCHEMA extractor;
```

- Create and prepare the schema for grafana data Connect to the grafana database and execute these commands, where you can provide your own name for the SCHEMA, and you grant privileges to the users and roles you created above:

```
CREATE SCHEMA grafana;
-/- read/write user has full access
GRANT ALL PRIVILEGES ON SCHEMA grafana TO grafana_backend;
-/- make sure the grafana user uses the grafana schema by default
ALTER ROLE grafana_backend SET search_path TO 'grafana';
```

The above commands provision the required databases and users. Your provisioning must be complete and correct for Embedded Reports to properly collect data and display it in Grafana reports and dashboards.

## Editing the CR File for Manually Created Databases

After you have provisioned the required database objects, you should edit the Turbonomic `cr.yaml` file to make use of these databases.

To edit the `cr.yaml` file:

1. Open the `.cr` file for editing.

The location of the file depends on the type of Turbonomic installation you are configuring:

- For an OVA installation of Turbonomic:

Open a SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

- Username: turbo
- Password: <your\_private\_password>

Then edit the file:

```
/opt/turbonomic/kubernetes/operator/deploy/crds/charts_v1alpha1_xl_cr.yaml
```

- Turbonomic on a Kubernetes node or node cluster:

Open the following file for editing:

```
deploy/crds/charts_v1alpha1_xl_cr.yaml
```

## 2. Specify the endpoint for connecting to the external database.

You can use the database service DNS, or you can use an IP address.

Add the endpoint as the `externalTimescaleDBIP` property in the `spec: global:` section of the `.cr` file:

```
spec:
 global:
 externalTimescaleDBIP: <host-or-IP>
```

## 3. Specify Turbonomic access the databases you provisioned.

As you specify users and databases, be sure to match the names you used above to provision users, databases, and schema. These instructions use the default naming that you can see above.

You will specify:

- Global read-only access to the extractor database `/spec/properties/global/dbs` This grants the extractor and the api components read-only access to the extractor database that you provisioned above. Those components will use the query user account.
- Read/write access to the extractor and grafana databases `/spec/properties/extractor/dbs` This grants the extractor component read/write access to the extractor and grafana databases that you provisioned above.
- Read/write access for Grafana to access the grafana database `/spec/grafana/grafana.ini/database`

Edit the `cr.yaml` file to add the following entries:

```
spec:
 properties:
 global:
 dbs:
 extractor:
 databaseName: extractor
 schemaName: extractor
 query:
 userName: query
 password: <default-password>
 extractor:
 dbs:
 extractor:
 userName: extractor
 password: <default-password>
 grafana:
 scehamName: grafana
 grafana:
 grafana.ini:
 database:
 name: grafana
 user: grafana_backend
 password: <default-password>
```

#### 4. (Recommended) Block automatic provisioning by Turbonomic on the TimescaleDB.

If you have manually provisioned the Embedded Reports objects on the TimescaleDB instance, then you should disable the options for Turbonomic to automatically execute provisioning on the TimescaleDB instance.

**NOTE:** that Turbonomic cannot provision on the TimescaleDB instance if you do not specify a global user account. If you performed these configuration steps, then you should not have done so. But we still recommend that you disable the options to execute provisioning.

Set the following properties to false:

- `shouldProvisionDatabase`: `spec/properties/global/dbs/postgresDefault/shouldProvisionDatabase` Determines whether Turbonomic will attempt to provision any database in scope of the the definition.
- `shouldProvisionUser`: `spec/properties/global/dbs/postgresDefault/shouldProvisionUser` Determines whether Turbonomic will attempt to provision any logins in scope of the definition.
- `destructiveProvisioningEnabled`: `spec/properties/global/dbs/postgresDefault/destructiveProvisioningEnabled` Determines whether, during its provisioning operations, Turbonomic can perform destructive operations like dropping databases, schemas or users that are found to be mis-configured. In the default installation scenario, the `public` schema that is created by PostgreSQL in any new database is dropped, to reduce the complexity of the overall model. Because this is considered a destructive operation, this option is `true` by default.

Edit the `cr.yaml` file to add the following properties:

```
spec:
 properties:
 global:
 dbs:
 postgresDefault:
 shouldProvisionDatabase: false
 shouldProvisionUser: false
 destructiveProvisioningEnabled: false
```

Placing these settings in the `/spec/properties/global/dbs/postgresDefault` block ensures that they will apply to all PostgreSQL databases and users used by any Turbonomic component. If your pre-provisioning is more selective than that, you can move the settings to more narrowly targeted property blocks like `/spec/properties/extractor/dbs/grafana` or `/spec/properties/global/dbs/extractor`.

## Automatic Provisioning of Users and Databases

For Turbonomic to automatically create the required databases for Embedded Reports, the Turbonomic `cr.yaml` file must specify credentials for a user account that has global R/W access to the DB service. To do this you can Store the credentials in cleartext in the `cr.yaml` file.

*If Automatic Provisioning is not appropriate for your environment*, then you must access the DB service with a R/W account and manually add the databases and user accounts that Embedded Reports can use. See [Manual Provisioning of Users and Databases](#).

To enable an external TimescaleDB for Embedded Reports, you must edit the Turbonomic `charts_v1alpha1_xl_cr.yaml` file.

#### 1. Open the `.cr` file for editing.

The location of the file depends on the type of Turbonomic installation you are configuring:

- For an OVA installation of Turbonomic:

Open a SSH terminal session on your Turbonomic instance.

Log in with the System Administrator that you set up when you installed Turbonomic:

– Username: `turbo`



– Password: <your\_private\_password>

Then edit the file:

/opt/turbonomic/kubernetes/operator/deploy/crds/charts\_v1alpha1\_x1\_cr.yaml

- Turbonomic on a Kubernetes node or node cluster:

Open the following file for editing:

deploy/crds/charts\_v1alpha1\_x1\_cr.yaml

2. Specify the endpoint for connecting to the external database.

For the connection endpoint, provide either the fully-qualified domain name of the external server or its IP address.

Add the endpoint to the spec/global/externalTimescaleDBIP property in the cr.yaml file:

```
spec:
 ...
 global:
 externalTimescaleDBIP: <host-or-IP>
```

3. Specify the global R/W account that Turbonomic will use to log into the TimescaleDB service.

This enables Turbonomic to automatically create the databases, schemas, and users that Embedded Reports requires.

**NOTE:** You should only do this if you want to enable global RW access to the TimescaleDB service in the cr.yaml file. You will specify the account credentials in clear text.

If you choose not to specify the global account here, then you must manually provision the database objects that Embedded Reports requires. See [Manual Provisioning of Users and Databases](#).

To grant the access Turbonomic needs, specify the username and password for an account that has global R/W privileges on the TimescaleDB instance. Add these credentials in the /spec/properties/global/dbs/postgresDefault block of the cr.yaml file, as follows:

```
spec:
 properties:
 global:
 dbs:
 postgresDefault:
 rootUserName: <root-user>
 rootPassword: <root-password>
```

## Reference: Working with YAML Files

**YAML** is the primary file format to create and configure resources on kubernetes, including everything to do with the Turbonomic platform. The Custom Resource YAML provides a convenient, single place to define the majority of configuration details for Turbonomic. When editing YAML files, you must be careful to respect the file syntax, including indents in the file. General rules for edits include:

- Always uses spaces, not tabs, for all indentation. Since spacing and indentations matter, and can yield an invalid result or a parameter completely skipped over, you should work with YAML files with an editor that supports using vertical lines associated with indentations to visually spot a misalignment. If your editor of choice makes this difficult, you can use the linux expand utility when you're done, to convert tabs to equivalent spaces.
- Indention uses two spaces per level.
- Be careful to keep the same indentation for all properties in a given section.
- Never use the same property name twice in the same section. Doing this will render the YAML file invalid, though you will likely not see any notification of a problem. Rather, all but one of the property definitions will be silently ignored.

## Spacing matters

For the Turbonomic Custom Resource, indentation defines where parameters are applied (globally or to specific components), so ensure you line up the text appropriately. The following example shows examples of specifications that are applied at a global level (`{ "spec": { "global": { "tag": "8.6.4" } } }`) which sets the container image tag for all instances. Then indented we see properties that are global for the remote database (`{ "spec": { "properties": { "global": { "dbPort": "6033" } } } }`) describes a property of `dbPort` that would be set for remote DB connections.

Each line is indented two spaces from the higher level.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

## Combine properties correctly

A YAML file is read top down, and if there are different parameters that apply to the same component, they need to be combined. The following example shows a YAML where properties for the `ui` component of image tag *and* memory limit resources.

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
 ui:
 resources:
 limits:
 memory: 4Gi
```

This YAML will not set both image tag and memory limit resources for the `ui` component because they are set in two different sections. The memory limits would be applied since that is the last section to be read, and it would overwrite the first set of properties with the image tag. The following YAML will set both image tag and memory limit resources for the `ui` component:

```
spec:
 global:
 repository: turbonomic
 tag: 8.6.4
 ui:
 image:
 tag: 8.0.5
 resources:
 limits:
 memory: 4Gi
 properties:
 global:
 dbPort: 6033
 kubeturbo:
 enabled: true
 aws:
 enabled: true
```

To avoid having the same property name twice in the same section, search for an existing instance of the key you want to add. With the complex embedded structure in many k8s resources, this can be quite challenging; however, you can use an online validator that does not accept duplicate keys to check your entire CR file (such as [YAMLint - The YAML Validator](#)).

## Finding a property

In this documentation we refer to specific properties in the CR file using a "path" expression. For example, assume the path `/spec/global/repository` to designate a `repository` property in the file. You can find the property in the file as follows:

1. Find a line that says `spec :` with no indentation at all.
2. Between that line and the next unindented line (not counting comments, which start with `#`), find a line that says `global :` and is at the next level of indentation.
3. Between that line and the next line with the same indentation, find a line that starts with `repository :`. That line is where the addressed property is defined.

In this example, find the `repository` property specified by `/spec/global/repository`:

```
apiVersion: charts.helm.k8s.io/v1alpha1
kind: X1
metadata:
 name: x1-release
spec:
 properties:
 global:
 repository: # This is NOT the correct property
 ...

 # Global settings
 global:
 repository: # This is the one we're after
```

The first `repository` property is not at `/spec/global/repository`, but at `/spec/properties/global/repository`.





