

IBM Cloud Pak AIOps 4.7.0 - PDF 2



Tables of Contents

APIs	1
Custom resource definitions	1
IBM Cloud Pak for AIOps CRDs	1
AIModelUI-v1	3
AIOpsEdge-v1beta1	5
AIOpsUI-v1	112
AIOpsUIExtension-v1	118
AIManager-v1beta1	119
Algorithm-v1alpha1	123
Algorithm-v1beta1	124
ASM-v1beta1.md	125
BaseUI-v1	128
BundleManifest-v1alpha1	130
BundleManifest-v1beta1	132
ConnectorUI-v1	133
InsightsUI-v1	135
Installation-v1alpha1	139
Tunnel-v1	141
Infrastructure Automation CRDs	144
IAConfig	145
IAManageService	146
IMInstall	152
Accessing APIs	156
Generating API keys for authentication	160
Integrated UI APIs	161
Creating custom home page cards with API	161
Identity Management (IM) APIs	172
Preparing to run API commands	172
OIDC Registration APIs	174
Identity Provider APIs	185
Issue resolution API	220
Probable cause customization API	222
Change risk API	226
Metric API	231
Automation (runbook) API	236
Topology APIs	238
Accessing topology service Swagger UI	239
Enabling access to URL routes	240
Topology API reference	241
Properties	242
Edge labels	245
Edge types	247
Entity types	250
Status (and state)	254
Timestamps	256
Topology examples	257
Virtual machine example	257

OpenStack example	260
Physical device example	268
Infrastructure Management APIs	272
API Overview	272
Prerequisites	272
REST API Authorization	273
Password Updates	274
CRUD	274
HTTP Basics	276
JSON Specification	281
Query Specification	286
Bulk Querying	290
Collection Filtering	291
Pagination	293
API Reference	294
Primary Collections	295
Actions	298
Alert Management	300
Authentication Management	302
Automate Model	306
Automate Domains	311
Automation Requests	312
Automate Workspaces	315
Categories and Tags	318
Chargebacks, Rates and Assignments	323
Cloud Volumes	355
Conditions	356
Configuration Script Payloads	358
Configuration Script Source Management	364
Custom Actions	367
Custom Attributes	373
Custom Button Management	376
Event Streams	380
Flavor Management	381
Generic Object Management	382
Group Management	391
Host Management	393
Instance Management	395
Metric Rollups	397
Network Routers	398
Notifications Support	399
Orchestration Support	403
Physical Server Management	406
Picture Management	408
Policy Management	409
Policy Collections	418
Provider Support	421
Provision Requests	431

Request Support	436
Reports Support	439
Role Management	448
Security Groups	451
Service Management	452
Service Queries	465
Service Custom Actions	468
Service Dialogs	475
Service Orders	478
Setting Ownership	482
Settings Management	484
Snapshot Management	486
Tags and Tagging	488
Tasks	495
Tenant Management	496
User Management	499
Virtual Machine Management	501
API Appendices	506
Quick Reference	506
Collection Metadata	508
Resource Attributes	511
Provider Types	522
Provision Request Supported Attributes	523
Advanced Provisioning Attributes	526
SOAP to REST Mapping guide	529
Deprecated Types	533
Deprecated Resource Attributes	533
API Examples	534
Foundation	534
Querying Examples	534
Paging Queries	536
Automation Requests	538
Create an Automation Request	538
Create multiple Automation Requests	540
Providers	543
Create a Provider	544
Create a Provider with multiple credentials	544
Update a Provider	545
Refresh a Provider	546
Delete a Provider	546
Provisioning Requests	548
Create a Provision Request	548
Create multiple Provision Requests	552
Service Catalogs	560
Add a Service Catalog	560
Add multiple Service Catalogs	561
Edit a Service Catalog	562
Edit Service Catalogs	562

Assign Service Templates	563
Unassign Service Templates	564
Delete Service Catalogs	565
Service Templates	565
Edit a Service Template	566
Edit Service Templates	566
Delete Service Templates	567
Services	568
Order a Service	568
Order Multiple Services	569
Edit a Service	571
Edit a Service with Patch Method	571
Edit a Service with PUT	572
Edit Services	572
Reconfigure a Service	573
Refresh Service Template Dialog Fields	574
Refresh Service Dialog Fields	576
Retire a Service	577
Retire Multiple Services	578
Trigger a Custom Action on a Service	580
Delete Services	584
VM Management	585
Start a VM	585
Stop a VM	586
Suspend a VM	586
Scan a VM	586
Set owner of a VM	587
Add Event to a VM	588
Delete a VM	588
Tag Management	589
Assign Tags	590
Assign Tags to Service Template	592
Unassign Tags	592
Unassign Tags from Service Template	594
Usage Flows	595
Compliance Policy Flow	595
Managed services APIs	607
API reference	607
Prerequisites	608
Common APIs	611
Provider and connection	612
Template and template instance	617
Terraform version	631
Terraform Provider Engine file APIs	633
Resource specific	634
Service and service instance	645
Custom action APIs	734
Dynamic parameter discovery	748

Applications	749
Shared parameter	756
SMTP connection	765
Usage specific	768
Example: Retrieve deployed templates list	769
Sample Python script	771
Content APIs	773
Upload cookbooks and roles	774
List cookbooks and roles	778
Clear cookbooks and roles	779
Get service logs	780

APIs

IBM Cloud Pak® for AIOps offers a set of tools and APIs that you can use to predict, communicate, and resolve events before they become serious problems for your organization.

Learn how to use the tools and APIs that work with IBM Cloud Pak for AIOps in the following topics:

- [Custom resource definitions](#)
- [Accessing APIs](#)
- [Integrated UI APIs](#)
- [User management \(IAM\) APIs](#)
- [licensing APIs](#)
- [Issue resolution API](#)
- [Probable Cause Customization API](#)
- [Change Risk API](#)
- [Metric API](#)
- [Automation \(runbook\) API](#)
- [Topology APIs](#)
- [Infrastructure Management API Guide](#)
- [Managed services APIs](#)

Important: You can directly access the Swagger UI for only the Topology API and Metric API.

- For more information about accessing Metric service Swagger UI, see [Accessing Metric service Swagger UI](#).
- For more information about accessing topology service Swagger UI, see [Accessing Topology service Swagger UI](#).

Custom resource definitions

Use the following information to learn about the custom resource definitions (CRDs) used by IBM Cloud Pak for AIOps and Infrastructure Automation.

For more information about API conventions and definitions, see [API Conventions](#) in the Kubernetes documentation.

- [IBM Cloud Pak for AIOps custom resource definitions](#)
- [Infrastructure Automation custom resource definitions](#)

IBM Cloud Pak for AIOps custom resource definitions

Use this information to learn about the custom resource definitions (CRDs) used by IBM Cloud Pak for AIOps.

In the following table, CRDs that do not have a hyperlink must not be manually edited unless otherwise instructed, and are manipulated only by the operator.

Table 1. IBM Cloud Pak for AIOps CRDs

Kind	API version	Operator name	Description
AIManagerMainProd	v1beta1	aimanager-operator	AIManagerMainProd is the Schema for the aimanagermainprods API. For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
AIManager	v1beta1	aimanager-operator	AIManager Deploys the IBM AIOps AI Manager service components of IBM Cloud Pak for AIOps. For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
Algorithm	v1alpha1	aimanager-operator	Algorithm is the Schema for the algorithms API
Algorithm	v1beta1	aimanager-operator	Algorithm is the Schema for the algorithms API
AIOpsEdge	v1beta1	aiopsedge-operator	AIOpsEdge is the Schema for the aiopsedges API
AIOpsKafkaTopic	v1beta1	aiopsedge-operator	AIOpsKafkaTopic is the Schema for the aiopskafkatopics API
AutomationAction	v1beta1	aiopsedge-operator	AutomationAction is the Schema for the automationactions API
BundleManifest	v1alpha1	aiopsedge-operator	BundleManifest is the Schema for the bundlemanifests API
BundleManifest	v1beta1	aiopsedge-operator	BundleManifest is the Schema for the bundlemanifests API
ConnectorComponent	v1beta1	aiopsedge-operator	ConnectorComponent is the Schema for the connectorcomponents API
ConnectorConfiguration	v1beta1	aiopsedge-operator	ConnectorConfiguration is the Schema for the connectorconfigurations API
ConnectorSchema	v1beta1	aiopsedge-operator	ConnectorSchema is the Schema for the connectorschemas API
GitApp	v1beta1	aiopsedge-operator	GitApp is the Schema for the GitApp API
MicroEdgeConfiguration	v1beta1	aiopsedge-operator	MicroEdgeConfiguration is the Schema for the microedgeconfigurations API
ASMFormation	v1beta1	asm-operator	Create a formation deployment. Documentation This product is not deployable standalone and it is part of the LICCR already accepted
ASM	v1beta1	asm-operator	(No Description)
AIOpsAnalyticsOrchestrator	v1beta1	ibm-aiops-ir-ai	AIOpsAnalyticsOrchestrator is the Schema for the aiopsanalyticsorchestrators API. This product is not deployable standalone and it is part of the LICCR already accepted
IssueResolutionCore	v1beta1	ibm-aiops-ir-core	IssueResolutionCore installs core components required for the issue resolution capabilities of IBM Cloud Pak for AIOps. This product is not deployable standalone and it is part of the LICCR already accepted.

Kind	API version	Operator name	Description
LifecycleService	v1beta1	ibm-aiops-ir-lifecycle	LifecycleService is the Schema for the lifecycle services API
LifecycleTrigger	v1beta1	ibm-aiops-ir-lifecycle	LifecycleTrigger is the Schema for the lifecycle triggers API
CustomSize	v1alpha1	ibm-aiops-orchestrator	CustomSize is the Schema for the custom sizes API. This API is an alpha feature that has not been made generally available.
Installation	v1alpha1	ibm-aiops-orchestrator	Leverage advanced, explainable AI to resolve incidents faster, gain visibility into critical workloads, and intelligently manage key application resources. Learn More . By installing this product you accept the license terms .
Tunnel	v1	ibm-secure-tunnel-operator	Tunnel is the Schema for the tunnels API
AIModelUI	v1	ibm-watson-aiops-ui-operator	For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
AIOpsUIExtension	v1	ibm-watson-aiops-ui-operator	AIOpsUIExtension is the Schema for the aiopsuiextensions API
AIOpsUI	v1	ibm-watson-aiops-ui-operator	For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
AIOpsUI is the Schema for the aiopsuis API.			
BaseUI	v1	ibm-watson-aiops-ui-operator	For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
ConnectorUI	v1	ibm-watson-aiops-ui-operator	For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .
InsightsUI	v1	ibm-watson-aiops-ui-operator	For documentation regarding install parameters, see IBM Documentation . By installing this product you accept the license terms .

AIModelUI/v1

For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#).

spec

Description: CommonSpecs - specs that are common to determining a deployment.

Type: object

Table 1. Description of AIModelUI/v1 spec

Property	Type	Description
spec	object	CommonSpecs - specs that are common to determining a deployment.
spec.configuration	object	Additional configuration for the UI deployment
spec.configuration.topologyModel	string	TopologyModel - Specify the terminology to use in CP4WAI Ops when defining collections of topology resource groups. Valid values are "application" or "service". If this setting is not specified, the default value will be "application".
spec.container	object	ContainerSpec - possible configurations around containers
spec.container.containerPort	integer	(No Description)
spec.container.env	array	(No Description)
spec.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.image	object	ImageSpec - specification around how to pull an image.
spec.image.path	string	(No Description)
spec.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.image.pullSecret	string	(No Description)
spec.license	object	License - license spec determines whether a license has been accepted
spec.license.accept	boolean	License acceptance switch
spec.name	string	Name of the service being deployed
spec.replicas	integer	(No Description)
spec.size	string	(No Description)
spec.tlsSecretName	string	(No Description)
spec.version	string	Version of the csv deploying
spec.zenserviceName	string	name of the zenservice cr that is used for this deployment

Description: (No Description)

Type: object

Table 1. Description of AIModelUI/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

AIOpsEdge/v1beta1

AIOpsEdge is the Schema for the aiopsedges API

spec

Description: AIOpsEdgeSpec defines the desired state of AIOpsEdge

Type: object

Table 1. Description of AIOpsEdge/v1beta1 spec

Property	Type	Description
spec	object	AIOpsEdgeSpec defines the desired state of AIOpsEdge
spec.certificates	object	Certificates contains information required to create and trust certificates for internal TLS communication.
spec.certificates.caConfigMapKey	string	Key in the ConfigMap identified by caConfigMapName that contains root certificates to be trusted for internal TLS.
spec.certificates.caConfigMapName	string	Name of the ConfigMap that contains root certificates to be trusted for internal TLS.
spec.certificates.issuer	string	Name of the cert-manager.io/Issuer that will be used to create TLS certificates.
spec.disableDefaultConnectors	boolean	DisableDefaultConnectors can be enabled to prevent the installation of default connectors
spec.elasticsearchSecret	string	The secret name for ElasticSearch. In future releases, OpenSearch will replace ElasticSearch, which is why this is optional. OpenSearch's secret does not depend on the installation name
spec.kafka	object	The specification of the Kafka and ZooKeeper clusters, and Topic Operator.
spec.kafka.clientsCa	object	Configuration of the clients certificate authority.

Property	Type	Description
spec.kafka.clientsCa.certificateExpirationPolicy	string	How should CA certificate expiration be handled when generateCertificateAuthority=true . The default is for a new CA certificate to be generated reusing the existing private key.
spec.kafka.clientsCa.generateCertificateAuthority	boolean	If true then Certificate Authority certificates will be generated automatically. Otherwise the user will need to provide a Secret with the CA certificate. Default is true.
spec.kafka.clientsCa.generateSecretOwnerReference	boolean	If true then the Certificate Authority certificates secrets owner reference is set to the Kafka object. Otherwise no owner reference is set so deleting the Kafka object won't delete the secrets. Default is true.
spec.kafka.clientsCa.renewalDays	integer	The number of days in the certificate renewal period. This is the number of days before the a certificate expires during which renewal actions may be performed. When generateCertificateAuthority is true, this will cause the generation of a new certificate. When generateCertificateAuthority is true, this will cause extra logging at WARN level about the pending certificate expiry. Default is 30.
spec.kafka.clientsCa.validityDays	integer	The number of days generated certificates should be valid for. The default is 365.
spec.kafka.clusterCa	object	Configuration of the cluster certificate authority.
spec.kafka.clusterCa.certificateExpirationPolicy	string	How should CA certificate expiration be handled when generateCertificateAuthority=true . The default is for a new CA certificate to be generated reusing the existing private key.
spec.kafka.clusterCa.generateCertificateAuthority	boolean	If true then Certificate Authority certificates will be generated automatically. Otherwise the user will need to provide a Secret with the CA certificate. Default is true.
spec.kafka.clusterCa.generateSecretOwnerReference	boolean	If true then the Certificate Authority certificates secrets owner reference is set to the Kafka object. Otherwise no owner reference is set so deleting the Kafka object won't delete the secrets. Default is true.
spec.kafka.clusterCa.renewalDays	integer	The number of days in the certificate renewal period. This is the number of days before the a certificate expires during which renewal actions may be performed. When generateCertificateAuthority is true, this will cause the generation of a new certificate. When generateCertificateAuthority is true, this will cause extra logging at WARN level about the pending certificate expiry. Default is 30.
spec.kafka.clusterCa.validityDays	integer	The number of days generated certificates should be valid for. The default is 365.
spec.kafka.cruiseControl	object	Configuration for Cruise Control deployment. Deploys a Cruise Control instance when specified.
spec.kafka.cruiseControl.brokerCapacity	object	The Cruise Control brokerCapacity configuration.

Property	Type	Description
spec.kafka.cruiseControl.brokerCapacity.cpuUtilization	integer	Broker capacity for CPU resource utilization as a percentage (0 - 100).
spec.kafka.cruiseControl.brokerCapacity.disk	string	Broker capacity for disk in bytes, for example, 100Gi.
spec.kafka.cruiseControl.brokerCapacity.inboundNetwork	string	Broker capacity for inbound network throughput in bytes per second, for example, 10000KB/s.
spec.kafka.cruiseControl.brokerCapacity.outboundNetwork	string	Broker capacity for outbound network throughput in bytes per second, for example 10000KB/s.
spec.kafka.cruiseControl.config	object	The Cruise Control configuration. For a full list of configuration options refer to https://github.com/linkedin/cruise-control/wiki/Configurations . Note that properties with the following prefixes cannot be set: bootstrap.servers, client.id, zookeeper., network., security., failed.brokers.zk.path, webserver.http., webserver.api.urlprefix, webserver.session.path, webserver.accesslog., two.step., request.reason.required, metric.reporter.sampler.bootstrap.servers, metric.reporter.topic, partition.metric.sample.store.topic, broker.metric.sample.store.topic, capacity.config.file, self.healing., anomaly.detection., ssl. (with the exception of: ssl.cipher.suites, ssl.protocol, ssl.enabled.protocols, webserver.http.cors.enabled, webserver.http.cors.origin, webserver.http.cors.exposeheaders).
spec.kafka.cruiseControl.image	string	The docker image for the pods.
spec.kafka.cruiseControl.jvmOptions	object	JVM Options for the Cruise Control container.
spec.kafka.cruiseControl.jvmOptions.-XX	object	A map of -XX options to the JVM.
spec.kafka.cruiseControl.jvmOptions.-Xms	string	-Xms option to to the JVM.
spec.kafka.cruiseControl.jvmOptions.-Xmx	string	-Xmx option to to the JVM.
spec.kafka.cruiseControl.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.cruiseControl.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the <code>-D</code> option to the JVM.

Property	Type	Description
spec.kafka.cruiseControl.livenessProbe	object	Pod liveness checking for the Cruise Control container.
spec.kafka.cruiseControl.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.cruiseControl.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.cruiseControl.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.cruiseControl.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.cruiseControl.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.cruiseControl.logging	object	Logging configuration (log4j1) for Cruise Control.
spec.kafka.cruiseControl.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.cruiseControl.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.cruiseControl.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.cruiseControl.metrics	object	The Prometheus JMX Exporter configuration. See https://github.com/prometheus/jmx_exporter for details of the structure of this configuration.
spec.kafka.cruiseControl.metricsConfig	object	Metrics configuration.
spec.kafka.cruiseControl.metricsConfig.type	string	Metrics type. Only 'jmxPrometheusExporter' supported currently.
spec.kafka.cruiseControl.metricsConfig.valueFrom	object	ConfigMap where the Prometheus JMX Exporter configuration is stored. For details of the structure of this configuration, see the {JMXExporter}.

Property	Type	Description
spec.kafka.cruiseControl.metricsConfig.valueFrom.configMapKeyRef	object	Reference to the key in the ConfigMap containing the metrics configuration.
spec.kafka.cruiseControl.metricsConfig.valueFrom.configMapKeyRef.key	string	Key corresponds to the JSON schema field "key".
spec.kafka.cruiseControl.metricsConfig.valueFrom.configMapKeyRef.name	string	Name corresponds to the JSON schema field "name".
spec.kafka.cruiseControl.metricsConfig.valueFrom.configMapKeyRef.optional	boolean	Optional corresponds to the JSON schema field "optional".
spec.kafka.cruiseControl.readinessProbe	object	Pod readiness checking for the Cruise Control container.
spec.kafka.cruiseControl.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.cruiseControl.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.cruiseControl.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.cruiseControl.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.cruiseControl.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.cruiseControl.resources	object	CPU and memory resources to reserve for the Cruise Control container.
spec.kafka.cruiseControl.resources.limits	object	Limits corresponds to the JSON schema field "limits".

Property	Type	Description
spec.kafka.cruiseControl.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.cruiseControl.template	object	Template to specify how Cruise Control resources, Deployments and Pods , are generated.
spec.kafka.cruiseControl.template.apiService	object	Template for Cruise Control API Service .
spec.kafka.cruiseControl.template.apiService.metadata	object	Metadata applied to the resource.
spec.kafka.cruiseControl.template.apiService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.apiService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.cruiseControlContainer	object	Template for the Cruise Control container.
spec.kafka.cruiseControl.template.cruiseControlContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext	object	Security context for the container.
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".

Property	Type	Description
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".

Property	Type	Description
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".

Property	Type	Description
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.cruiseControl.template.cruiseControlContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.cruiseControl.template.deployment	object	Template for Cruise Control Deployment .
spec.kafka.cruiseControl.template.deployment.metadata	object	Metadata applied to the resource.
spec.kafka.cruiseControl.template.deployment.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.deployment.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.pod	object	Template for Cruise Control Pods .
spec.kafka.cruiseControl.template.pod.affinity	object	The pod's affinity rules.

Property	Type	Description
spec.kafka.cruiseControl.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.cruiseControl.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.cruiseControl.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.cruiseControl.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".

Property	Type	Description
spec.kafka.cruiseControl.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.cruiseControl.template.pod.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
spec.kafka.cruiseControl.template.pod.imagePullSecrets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.cruiseControl.template.pod.metadata	object	Metadata applied to the resource.
spec.kafka.cruiseControl.template.pod.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.pod.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.pod.priorityClassName	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.
spec.kafka.cruiseControl.template.pod.schedulerName	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.cruiseControl.template.pod.securityContext	object	Configures pod-level security attributes and common container settings.

Property	Type	Description
spec.kafka.cruiseControl.template.pod.securityContext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".
spec.kafka.cruiseControl.template.pod.securityContext.fsGroupChangePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.cruiseControl.template.pod.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.cruiseControl.template.pod.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.cruiseControl.template.pod.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.cruiseControl.template.pod.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.cruiseControl.template.pod.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.cruiseControl.template.pod.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.cruiseControl.template.pod.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.cruiseControl.template.pod.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".

Property	Type	Description
spec.kafka.cruiseControl.template.pod.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.cruiseControl.template.pod.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.cruiseControl.template.pod.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.cruiseControl.template.pod.securityContext.supplementalGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".
spec.kafka.cruiseControl.template.pod.securityContext.sysctls	array	Sysctls corresponds to the JSON schema field "sysctls".
spec.kafka.cruiseControl.template.pod.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.cruiseControl.template.pod.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.cruiseControl.template.pod.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.cruiseControl.template.pod.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".

Property	Type	Description
spec.kafka.cruiseControl.template.pod.terminationGracePeriodSeconds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.cruiseControl.template.pod.tolerations	array	The pod's tolerations.
spec.kafka.cruiseControl.template.pod.topologySpreadConstraints	array	The pod's topology spread constraints.
spec.kafka.cruiseControl.template.podDisruptionBudget	object	Template for Cruise Control PodDisruptionBudget .
spec.kafka.cruiseControl.template.podDisruptionBudget.maxUnavailable	integer	Maximum number of unavailable pods to allow automatic Pod eviction. A Pod eviction is allowed when the maxUnavailable number of pods or fewer are unavailable after the eviction. Setting this value to 0 prevents all voluntary evictions, so the pods must be evicted manually. Defaults to 1.
spec.kafka.cruiseControl.template.podDisruptionBudget.metadata	object	Metadata to apply to the PodDisruptionBudgetTemplate resource.
spec.kafka.cruiseControl.template.podDisruptionBudget.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.podDisruptionBudget.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.cruiseControl.template.tlsSidecarContainer	object	Template for the Cruise Control TLS sidecar container.
spec.kafka.cruiseControl.template.tlsSidecarContainer.env	array	Environment variables which should be applied to the container.

Property	Type	Description
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext	object	Security context for the container.
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".

Property	Type	Description
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".

Property	Type	Description
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.cruiseControl.template.tlsSidecarContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.cruiseControl.tlsSidecar	object	TLS sidecar configuration.

Property	Type	Description
spec.kafka.cruiseControl.tlsSidecar.image	string	The docker image for the container.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe	object	Pod liveness checking.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.logLevel	string	The log level for the TLS sidecar. Default value is notice .
spec.kafka.cruiseControl.tlsSidecar.readinessProbe	object	Pod readiness checking.
spec.kafka.cruiseControl.tlsSidecar.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.

Property	Type	Description
spec.kafka.cruiseControl.tlsSidecar.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.cruiseControl.tlsSidecar.resources	object	CPU and memory resources to reserve.
spec.kafka.cruiseControl.tlsSidecar.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.cruiseControl.tlsSidecar.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.entityOperator	object	Configuration of the Entity Operator.
spec.kafka.entityOperator.affinity	object	The pod's affinity rules.
spec.kafka.entityOperator.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.entityOperator.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".

Property	Type	Description
spec.kafka.entityOperator.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.entityOperator.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.entityOperator.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.entityOperator.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template	object	Template for Entity Operator resources. The template allows users to specify how is the Deployment and Pods generated.
spec.kafka.entityOperator.template.deployment	object	Template for Entity Operator Deployment .
spec.kafka.entityOperator.template.deployment.metadata	object	Metadata applied to the resource.

Property	Type	Description
spec.kafka.entityOperator.template.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.entityOperator.template.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.entityOperator.template.pod	object	Template for Entity Operator Pods .
spec.kafka.entityOperator.template.pod.affinity	object	The pod's affinity rules.
spec.kafka.entityOperator.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.entityOperator.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.entityOperator.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".

Property	Type	Description
spec.kafka.entityOperator.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.entityOperator.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.entityOperator.template.pod.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
spec.kafka.entityOperator.template.pod.imagePullSecrets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.entityOperator.template.pod.metadata	object	Metadata applied to the resource.

Property	Type	Description
spec.kafka.entityOperator.template.pod.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.entityOperator.template.pod.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.entityOperator.template.pod.priorityClassName	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.
spec.kafka.entityOperator.template.pod.schedulerName	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.entityOperator.template.pod.securityContext	object	Configures pod-level security attributes and common container settings.
spec.kafka.entityOperator.template.pod.securityContext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".
spec.kafka.entityOperator.template.pod.securityContext.fsGroupChangePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.entityOperator.template.pod.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.entityOperator.template.pod.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.entityOperator.template.pod.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.entityOperator.template.pod.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".

Property	Type	Description
spec.kafka.entityOperator.template.pod.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.entityOperator.template.pod.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.entityOperator.template.pod.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.pod.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.entityOperator.template.pod.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.entityOperator.template.pod.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.entityOperator.template.pod.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.pod.securityContext.supplementalGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".
spec.kafka.entityOperator.template.pod.securityContext.sysctls	array	Sysctls corresponds to the JSON schema field "sysctls".
spec.kafka.entityOperator.template.pod.securityContext.windowOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".

Property	Type	Description
spec.kafka.entityOperator.template.pod.securityContext.windowOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.entityOperator.template.pod.securityContext.windowOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.entityOperator.template.pod.securityContext.windowOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.entityOperator.template.pod.terminationGracePeriodSeconds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.entityOperator.template.pod.tolerations	array	The pod's tolerations.
spec.kafka.entityOperator.template.pod.topologySpreadConstraints	array	The pod's topology spread constraints.
spec.kafka.entityOperator.template.tlsSidecarContainer	object	Template for the Entity Operator TLS sidecar container.
spec.kafka.entityOperator.template.tlsSidecarContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext	object	Security context for the container.

Property	Type	Description
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".

Property	Type	Description
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".

Property	Type	Description
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.windowOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.windowOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.windowOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.entityOperator.template.tlsSidecarContainer.securityContext.windowOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.entityOperator.template.topicOperatorContainer	object	Template for the Entity Topic Operator container.
spec.kafka.entityOperator.template.topicOperatorContainer.env	array	Environment variables which should be applied to the container.

Property	Type	Description
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext	object	Security context for the container.
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".

Property	Type	Description
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".

Property	Type	Description
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".

Property	Type	Description
spec.kafka.entityOperator.template.topicOperatorContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.entityOperator.template.userOperatorContainer	object	Template for the Entity User Operator container.
spec.kafka.entityOperator.template.userOperatorContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.entityOperator.template.userOperatorContainer.securityContext	object	Security context for the container.
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".

Property	Type	Description
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".

Property	Type	Description
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".

Property	Type	Description
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.entityOperator.template.userOperatorContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.entityOperator.tlsSidecar	object	TLS sidecar configuration.
spec.kafka.entityOperator.tlsSidecar.image	string	The docker image for the container.
spec.kafka.entityOperator.tlsSidecar.livenessProbe	object	Pod liveness checking.
spec.kafka.entityOperator.tlsSidecar.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.tlsSidecar.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.

Property	Type	Description
spec.kafka.entityOperator.tlsSidecar.logLevel	string	The log level for the TLS sidecar. Default value is notice .
spec.kafka.entityOperator.tlsSidecar.readinessProbe	object	Pod readiness checking.
spec.kafka.entityOperator.tlsSidecar.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.tlsSidecar.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.entityOperator.tlsSidecar.resources	object	CPU and memory resources to reserve.
spec.kafka.entityOperator.tlsSidecar.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.entityOperator.tlsSidecar.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.entityOperator.tolerations	array	The pod's tolerations.
spec.kafka.entityOperator.topicOperator	object	Configuration of the Topic Operator.

Property	Type	Description
spec.kafka.entityOperator.topicOperator.image	string	The image to use for the Topic Operator.
spec.kafka.entityOperator.topicOperator.jvmOptions	object	JVM Options for pods.
spec.kafka.entityOperator.topicOperator.jvmOptions.-XX	object	A map of -XX options to the JVM.
spec.kafka.entityOperator.topicOperator.jvmOptions.-Xms	string	-Xms option to to the JVM.
spec.kafka.entityOperator.topicOperator.jvmOptions.-Xmx	string	-Xmx option to to the JVM.
spec.kafka.entityOperator.topicOperator.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.entityOperator.topicOperator.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the <code>-D</code> option to the JVM.
spec.kafka.entityOperator.topicOperator.livenessProbe	object	Pod liveness checking.
spec.kafka.entityOperator.topicOperator.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.topicOperator.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.

Property	Type	Description
spec.kafka.entityOperator.topicOperator.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.logging	object	Logging configuration.
spec.kafka.entityOperator.topicOperator.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.entityOperator.topicOperator.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.entityOperator.topicOperator.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.entityOperator.topicOperator.readinessProbe	object	Pod readiness checking.
spec.kafka.entityOperator.topicOperator.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.topicOperator.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.

Property	Type	Description
spec.kafka.entityOperator.topicOperator.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.entityOperator.topicOperator.reconciliationIntervalSeconds	integer	Interval between periodic reconciliations.
spec.kafka.entityOperator.topicOperator.resources	object	CPU and memory resources to reserve.
spec.kafka.entityOperator.topicOperator.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.entityOperator.topicOperator.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.entityOperator.topicOperator.topicMetadataMaxAttempts	integer	The number of attempts at getting topic metadata.
spec.kafka.entityOperator.topicOperator.watchedNamespace	string	The namespace the Topic Operator should watch.
spec.kafka.entityOperator.topicOperator.zookeeperSessionTimeoutSeconds	integer	Timeout for the ZooKeeper session.
spec.kafka.entityOperator.userOperator	object	Configuration of the User Operator.
spec.kafka.entityOperator.userOperator.image	string	The image to use for the User Operator.
spec.kafka.entityOperator.userOperator.jvmOptions	object	JVM Options for pods.
spec.kafka.entityOperator.userOperator.jvmOptions.-XX	object	A map of -XX options to the JVM.

Property	Type	Description
spec.kafka.entityOperator.jvmOptions.-Xms	string	-Xms option to the JVM.
spec.kafka.entityOperator.jvmOptions.-Xmx	string	-Xmx option to the JVM.
spec.kafka.entityOperator.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.entityOperator.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the <code>-D</code> option to the JVM.
spec.kafka.entityOperator.livenessProbe	object	Pod liveness checking.
spec.kafka.entityOperator.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.entityOperator.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.entityOperator.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.entityOperator.logging	object	Logging configuration.

Property	Type	Description
spec.kafka.entityOperator.userOperator.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.entityOperator.userOperator.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.entityOperator.userOperator.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.entityOperator.readinessProbe	object	Pod readiness checking.
spec.kafka.entityOperator.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.entityOperator.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.entityOperator.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.entityOperator.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.entityOperator.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.entityOperator.reconciliationIntervalSeconds	integer	Interval between periodic reconciliations.
spec.kafka.entityOperator.resources	object	CPU and memory resources to reserve.

Property	Type	Description
spec.kafka.entityOperator.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.entityOperator.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.entityOperator.secretPrefix	string	The prefix that will be added to the KafkaUser name to be used as the Secret name.
spec.kafka.entityOperator.watchedNamespace	string	The namespace the User Operator should watch.
spec.kafka.entityOperator.zookeeperSessionTimeoutSeconds	integer	Timeout for the ZooKeeper session.
spec.kafka.jmxTrans	object	Configuration for JmxTrans. When the property is present a JmxTrans deployment is created for gathering JMX metrics from each Kafka broker. For more information see https://github.com/jmxtrans/jmxtrans [JmxTrans GitHub].
spec.kafka.jmxTrans.image	string	The image to use for the JmxTrans.
spec.kafka.jmxTrans.kafkaQueries	array	Queries to send to the Kafka brokers to define what data should be read from each broker. For more information on these properties see, xref:type-JmxTransQueryTemplate-reference[JmxTransQueryTemplate schema reference].
spec.kafka.jmxTrans.logLevel	string	Sets the logging level of the JmxTrans deployment. For more information see, https://github.com/jmxtrans/jmxtrans-agent/wiki/Troubleshooting [JmxTrans Logging Level].
spec.kafka.jmxTrans.outputDefinitions	array	Defines the output hosts that will be referenced later on. For more information on these properties see, xref:type-JmxTransOutputDefinitionTemplate-reference[JmxTransOutputDefinitionTemplate schema reference].
spec.kafka.jmxTrans.resources	object	CPU and memory resources to reserve.
spec.kafka.jmxTrans.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.jmxTrans.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.jmxTrans.template	object	Template for JmxTrans resources.

Property	Type	Description
spec.kafka.jmxTran.template.container	object	Template for JmxTrans container.
spec.kafka.jmxTran.template.container.env	array	Environment variables which should be applied to the container.
spec.kafka.jmxTran.template.container.securityContext	object	Security context for the container.
spec.kafka.jmxTran.template.container.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.jmxTran.template.container.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.jmxTran.template.container.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.jmxTran.template.container.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.jmxTran.template.container.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.jmxTran.template.container.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.jmxTran.template.container.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".

Property	Type	Description
spec.kafka.jmxT rans.template.c ontainer.securit yContext.runAs Group	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.runAs NonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.runAs User	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.seLinu xOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.seLinu xOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.seLinu xOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.seLinu xOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.seLinu xOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.jmxT rans.template.c ontainer.securit yContext.secco mpProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".

Property	Type	Description
spec.kafka.jmxTran.template.container.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.jmxTran.template.container.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.jmxTran.template.container.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.jmxTran.template.container.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.jmxTran.template.container.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.jmxTran.template.container.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.jmxTran.template.deployment	object	Template for JmxTrans Deployment .
spec.kafka.jmxTran.template.deployment.metadata	object	Metadata applied to the resource.
spec.kafka.jmxTran.template.deployment.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .

Property	Type	Description
spec.kafka.jmxTrans.template.deployment.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.jmxTrans.template.pod	object	Template for JmxTrans Pods .
spec.kafka.jmxTrans.template.pod.affinity	object	The pod's affinity rules.
spec.kafka.jmxTrans.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.jmxTrans.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.jmxTrans.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.jmxTrans.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.jmxTrans.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.jmxTrans.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".

Property	Type	Description
spec.kafka.jmxT rans.template.p od.affinity.podA ffinity.requiredDu ringSchedulingI gnoredDuringEx ecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.jmxT rans.template.p od.affinity.podA ntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.jmxT rans.template.p od.affinity.podA ntiAffinity.prefer redDuringSched ulingIgnoredDur ingExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.jmxT rans.template.p od.affinity.podA ntiAffinity.require dDuringSchedu lingIgnoredDuri ngExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.jmxT rans.template.p od.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
spec.kafka.jmxT rans.template.p od.imagePullSec rets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.jmxT rans.template.p od.metadata	object	Metadata applied to the resource.
spec.kafka.jmxT rans.template.p od.metadata.an notations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.jmxT rans.template.p od.metadata.lab els	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.jmxT rans.template.p od.priorityClass Name	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.

Property	Type	Description
spec.kafka.jmxT rans.template.p od.schedulerNa me	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.jmxT rans.template.p od.securityCont ext	object	Configures pod-level security attributes and common container settings.
spec.kafka.jmxT rans.template.p od.securityCont ext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".
spec.kafka.jmxT rans.template.p od.securityCont ext.fsGroupCha ngePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.jmxT rans.template.p od.securityCont ext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.jmxT rans.template.p od.securityCont ext.runAsNonRo ot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.jmxT rans.template.p od.securityCont ext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.jmxT rans.template.p od.securityCont ext.seLinuxOpti ons	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.jmxT rans.template.p od.securityCont ext.seLinuxOpti ons.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.jmxT rans.template.p od.securityCont ext.seLinuxOpti ons.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.jmxT rans.template.p od.securityCont ext.seLinuxOpti ons.type	string	Type corresponds to the JSON schema field "type".

Property	Type	Description
spec.kafka.jmxT rans.template.p od.securityCont ext.seLinuxOpti ons.user	string	User corresponds to the JSON schema field "user".
spec.kafka.jmxT rans.template.p od.securityCont ext.seccompPro file	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.jmxT rans.template.p od.securityCont ext.seccompPro file.localhostPro file	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.jmxT rans.template.p od.securityCont ext.seccompPro file.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.jmxT rans.template.p od.securityCont ext.supplement alGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".
spec.kafka.jmxT rans.template.p od.securityCont ext.sysctls	array	Sysctls corresponds to the JSON schema field "sysctls".
spec.kafka.jmxT rans.template.p od.securityCont ext.windowsOpti ons	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.jmxT rans.template.p od.securityCont ext.windowsOpti ons.gmsaCrede ntialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.jmxT rans.template.p od.securityCont ext.windowsOpti ons.gmsaCrede ntialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".

Property	Type	Description
spec.kafka.jmxT rans.template.p od.securityCont ext.windowsOpti ons.runAsUserN ame	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.jmxT rans.template.p od.terminationG racePeriodSeco nds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.jmxT rans.template.p od.tolerations	array	The pod's tolerations.
spec.kafka.jmxT rans.template.p od.topologySpre adConstraints	array	The pod's topology spread constraints.
spec.kafka.kafk a	object	Configuration of the Kafka cluster.
spec.kafka.kafk a.affinity	object	The pod's affinity rules.
spec.kafka.kafk a.affinity.nodeAf finity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.kafk a.affinity.nodeAf finity.preferredD uringScheduling IgnoredDuringE xecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafk a.affinity.nodeAf finity.requiredDu ringSchedulingI gnoredDuringEx ecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafk a.affinity.nodeAf finity.requiredDu ringSchedulingI gnoredDuringEx ecution.nodeSel ectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.kafk a.affinity.podAffi nity	object	PodAffinity corresponds to the JSON schema field "podAffinity".

Property	Type	Description
spec.kafka.kafka.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.kafka.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.authorization	object	Authorization configuration for Kafka brokers.
spec.kafka.kafka.authorization.allowOnError	boolean	Defines whether a Kafka client should be allowed or denied by default when the authorizer fails to query the Open Policy Agent, for example, when it is temporarily unavailable). Defaults to false - all actions will be denied.
spec.kafka.kafka.authorization.clientId	string	OAuth Client ID which the Kafka client can use to authenticate against the OAuth server and use the token endpoint URI.
spec.kafka.kafka.authorization.delegateToKafkaAcls	boolean	Whether authorization decision should be delegated to the 'Simple' authorizer if DENIED by Keycloak Authorization Services policies. Default value is false .
spec.kafka.kafka.authorization.disableTlsHostnameVerification	boolean	Enable or disable TLS hostname verification. Default value is false .
spec.kafka.kafka.authorization.expireAfterMs	integer	The expiration of the records kept in the local cache to avoid querying the Open Policy Agent for every request. Defines how often the cached authorization decisions are reloaded from the Open Policy Agent server. In milliseconds. Defaults to 3600000 .

Property	Type	Description
spec.kafka.kafka.authorization.grantsRefreshPeriodSeconds	integer	The time between two consecutive grants refresh runs in seconds. The default value is 60.
spec.kafka.kafka.authorization.grantsRefreshPoolSize	integer	The number of threads to use to refresh grants for active sessions. The more threads, the more parallelism, so the sooner the job completes. However, using more threads places a heavier load on the authorization server. The default value is 5.
spec.kafka.kafka.authorization.initialCacheCapacity	integer	Initial capacity of the local cache used by the authorizer to avoid querying the Open Policy Agent for every request Defaults to 5000 .
spec.kafka.kafka.authorization.maximumCacheSize	integer	Maximum capacity of the local cache used by the authorizer to avoid querying the Open Policy Agent for every request. Defaults to 50000 .
spec.kafka.kafka.authorization.superUsers	array	List of super users. Should contain list of user principals which should get unlimited access rights.
spec.kafka.kafka.authorization.tlsTrustedCertificates	array	Trusted certificates for TLS connection to the OAuth server.
spec.kafka.kafka.authorization.tokenEndpointUri	string	Authorization server token endpoint URI.
spec.kafka.kafka.authorization.type	string	Authorization type. Currently, the supported types are simple , keycloak , and opa . simple authorization type uses Kafka's kafka.security.authorizer.AclAuthorizer class for authorization. keycloak authorization type uses Keycloak Authorization Services for authorization. opa authorization type uses Open Policy Agent based authorization.
spec.kafka.kafka.authorization.url	string	The URL used to connect to the Open Policy Agent server. The URL has to include the policy which will be queried by the authorizer. This option is required.
spec.kafka.kafka.brokerRackInitImage	string	The image of the init container used for initializing the broker.rack .

Property	Type	Description
spec.kafka.kafka.config	object	Kafka broker config properties with the following prefixes cannot be set: listeners, advertised., broker., listener., host.name, port, inter.broker.listener.name, sasl., ssl., security., password., principal.builder.class, log.dir, zookeeper.connect, zookeeper.set.acl, zookeeper.ssl, zookeeper.clientCnxnSocket, authorizer., super.user, cruise.control.metrics.topic, cruise.control.metrics.reporter.bootstrap.servers (with the exception of: zookeeper.connection.timeout.ms, ssl.cipher.suites, ssl.protocol, ssl.enabled.protocols, cruise.control.metrics.topic.num.partitions, cruise.control.metrics.topic.replication.factor, cruise.control.metrics.topic.retention.ms, cruise.control.metrics.topic.auto.create.retries, cruise.control.metrics.topic.auto.create.timeout.ms, cruise.control.metrics.topic.min.insync.replicas).
spec.kafka.kafka.image	string	The docker image for the pods. The default value depends on the configured Kafka.spec.kafka.version .
spec.kafka.kafka.jmxOptions	object	JMX Options for Kafka brokers.
spec.kafka.kafka.jmxOptions.authentication	object	Authentication configuration for connecting to the Kafka JMX port.
spec.kafka.kafka.jmxOptions.authentication.type	string	Authentication type. Currently the only supported types are password.password type creates a username and protected port with no TLS.
spec.kafka.kafka.jvmOptions	object	JVM Options for pods.
spec.kafka.kafka.jvmOptions.-XX	object	A map of -XX options to the JVM.
spec.kafka.kafka.jvmOptions.-Xms	string	-Xms option to to the JVM.
spec.kafka.kafka.jvmOptions.-Xmx	string	-Xmx option to to the JVM.
spec.kafka.kafka.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.kafka.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the -D option to the JVM.
spec.kafka.kafka.listeners	array	Listeners corresponds to the JSON schema field "listeners".
spec.kafka.kafka.livenessProbe	object	Pod liveness checking.

Property	Type	Description
spec.kafka.kafka.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.kafka.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.kafka.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafka.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafka.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafka.logging	object	Logging configuration for Kafka.
spec.kafka.kafka.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.kafka.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.kafka.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.kafka.metrics	object	The Prometheus JMX Exporter configuration. See https://github.com/prometheus/jmx_exporter for details of the structure of this configuration.
spec.kafka.kafka.metricsConfig	object	Metrics configuration.
spec.kafka.kafka.metricsConfig.type	string	Metrics type. Only 'jmxPrometheusExporter' supported currently.
spec.kafka.kafka.metricsConfig.valueFrom	object	ConfigMap where the Prometheus JMX Exporter configuration is stored. For details of the structure of this configuration, see the {JMXExporter}.
spec.kafka.kafka.metricsConfig.valueFrom.configMapKeyRef	object	Reference to the key in the ConfigMap containing the metrics configuration.
spec.kafka.kafka.metricsConfig.valueFrom.configMapKeyRef.key	string	Key corresponds to the JSON schema field "key".

Property	Type	Description
spec.kafka.kafka.metricsConfig.valueFrom.configMapKeyRef.name	string	Name corresponds to the JSON schema field "name".
spec.kafka.kafka.metricsConfig.valueFrom.configMapKeyRef.optional	boolean	Optional corresponds to the JSON schema field "optional".
spec.kafka.kafka.rack	object	Configuration of the broker.rack broker config.
spec.kafka.kafka.rack.topologyKey	string	A key that matches labels assigned to the Kubernetes cluster nodes. The value of the label is used to set the broker's broker.rack config and client.rack in Kafka Connect.
spec.kafka.kafka.readinessProbe	object	Pod readiness checking.
spec.kafka.kafka.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.kafka.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.kafka.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafka.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafka.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafka.replicas	integer	The number of pods in the cluster.
spec.kafka.kafka.resources	object	CPU and memory resources to reserve.
spec.kafka.kafka.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.kafka.resources.requests	object	Requests corresponds to the JSON schema field "requests".

Property	Type	Description
spec.kafka.kafka.storage	object	Storage configuration (disk). Cannot be updated.
spec.kafka.kafka.storage.class	string	The storage class to use for dynamic volume allocation.
spec.kafka.kafka.storage.deleteClaim	boolean	Specifies if the persistent volume claim has to be deleted when the cluster is un-deployed.
spec.kafka.kafka.storage.id	integer	Storage identification number. It is mandatory only for storage volumes defined in a storage of type 'jbod'.
spec.kafka.kafka.storage.overrides	array	Overrides for individual brokers. The overrides field allows to specify a different configuration for different brokers.
spec.kafka.kafka.storage.selector	object	Specifies a specific persistent volume to use. It contains key:value pairs representing labels for selecting such a volume.
spec.kafka.kafka.storage.size	string	When type=persistent-claim, defines the size of the persistent volume claim (i.e 1Gi). Mandatory when type=persistent-claim.
spec.kafka.kafka.storage.sizeLimit	string	When type=ephemeral, defines the total amount of local storage required for this EmptyDir volume (for example 1Gi).
spec.kafka.kafka.storage.type	string	Storage type, must be either 'ephemeral', 'persistent-claim', or 'jbod'.
spec.kafka.kafka.storage.volumes	array	List of volumes as Storage objects representing the JBOD disks array.
spec.kafka.kafka.template	object	Template for Kafka cluster resources. The template allows users to specify how are the StatefulSet , Pods and Services generated.
spec.kafka.kafka.template.bootstrapService	object	Template for Kafka bootstrap Service .
spec.kafka.kafka.template.bootstrapService.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.bootstrapService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.bootstrapService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.brokersService	object	Template for Kafka broker Service .

Property	Type	Description
spec.kafka.kafka.template.brokersService.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.brokersService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.brokersService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.clusterCaCert	object	Template for Secret with Kafka Cluster certificate public key.
spec.kafka.kafka.template.clusterCaCert.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.clusterCaCert.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.clusterCaCert.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapIngress	object	Template for Kafka external bootstrap Ingress .
spec.kafka.kafka.template.externalBootstrapIngress.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.externalBootstrapIngress.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapIngress.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapRoute	object	Template for Kafka external bootstrap Route .

Property	Type	Description
spec.kafka.kafka.template.externalBootstrapRoute.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.externalBootstrapRoute.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapRoute.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapService	object	Template for Kafka external bootstrap Service .
spec.kafka.kafka.template.externalBootstrapService.externalTrafficPolicy	string	Specifies whether the service routes external traffic to node-local or cluster-wide endpoints. Cluster may cause a second hop to another node and obscures the client source IP. Local avoids a second hop for LoadBalancer and Nodeport type services and preserves the client source IP (when supported by the infrastructure). If unspecified, Kubernetes will use Cluster as the default.
spec.kafka.kafka.template.externalBootstrapService.loadBalancerSourceRanges	array	A list of CIDR ranges (for example 10.0.0.0/8 or 130.211.204.1/32) from which clients can connect to load balancer type listeners. If supported by the platform, traffic through the loadbalancer is restricted to the specified CIDR ranges. This field is applicable only for loadbalancer type services and is ignored if the cloud provider does not support the feature. For more information, see https://v1-17.docs.kubernetes.io/docs/tasks/access-application-cluster/configure-cloud-provider-firewall/ .
spec.kafka.kafka.template.externalBootstrapService.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.externalBootstrapService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.externalBootstrapService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.initContainer	object	Template for the Kafka init container.

Property	Type	Description
spec.kafka.kafka.template.initContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.kafka.template.initContainer.securityContext	object	Security context for the container.
spec.kafka.kafka.template.initContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.kafka.template.initContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.kafka.template.initContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.kafka.template.initContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.kafka.template.initContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.kafka.template.initContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.kafka.template.initContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.kafka.template.initContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".

Property	Type	Description
spec.kafka.kafka.template.initContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafka.template.initContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.kafka.template.initContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.kafka.template.initContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafka.template.initContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafka.template.initContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.initContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.kafka.template.initContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafka.template.initContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".

Property	Type	Description
spec.kafka.kafka.template.initContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.initContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafka.template.initContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.kafka.template.initContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafka.template.initContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.kafka.template.kafkaContainer	object	Template for the Kafka broker container.
spec.kafka.kafka.template.kafkaContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.kafka.template.kafkaContainer.securityContext	object	Security context for the container.
spec.kafka.kafka.template.kafkaContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.kafka.template.kafkaContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".

Property	Type	Description
spec.kafka.kafka.template.kafkaContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.kafka.template.kafkaContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.kafka.template.kafkaContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.kafka.template.kafkaContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.kafka.template.kafkaContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.kafka.template.kafkaContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.kafka.template.kafkaContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafka.template.kafkaContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.kafka.template.kafkaContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".

Property	Type	Description
spec.kafka.kafka.template.kafkaContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafka.template.kafkaContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafka.template.kafkaContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.kafkaContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.kafka.template.kafkaContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafka.template.kafkaContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.kafka.template.kafkaContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.kafkaContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafka.template.kafkaContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".

Property	Type	Description
spec.kafka.kafka.template.kafkaContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafka.template.kafkaContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.kafka.template.perPodIngress	object	Template for Kafka per-pod Ingress used for access from outside of Kubernetes.
spec.kafka.kafka.template.perPodIngress.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.perPodIngress.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.perPodIngress.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.perPodRoute	object	Template for Kafka per-pod Routes used for access from outside of OpenShift.
spec.kafka.kafka.template.perPodRoute.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.perPodRoute.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.perPodRoute.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.perPodService	object	Template for Kafka per-pod Services used for access from outside of Kubernetes.

Property	Type	Description
spec.kafka.kafka.template.perPodService.externalTrafficPolicy	string	Specifies whether the service routes external traffic to node-local or cluster-wide endpoints. Cluster may cause a second hop to another node and obscures the client source IP. Local avoids a second hop for LoadBalancer and Nodeport type services and preserves the client source IP (when supported by the infrastructure). If unspecified, Kubernetes will use Cluster as the default.
spec.kafka.kafka.template.perPodService.loadBalancerSourceRanges	array	A list of CIDR ranges (for example 10.0.0.0/8 or 130.211.204.1/32) from which clients can connect to load balancer type listeners. If supported by the platform, traffic through the loadbalancer is restricted to the specified CIDR ranges. This field is applicable only for loadbalancer type services and is ignored if the cloud provider does not support the feature. For more information, see https://v1-17.docs.kubernetes.io/docs/tasks/access-application-cluster/configure-cloud-provider-firewall/ .
spec.kafka.kafka.template.perPodService.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.perPodService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.perPodService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.persistentVolumeClaim	object	Template for all Kafka PersistentVolumeClaims .
spec.kafka.kafka.template.persistentVolumeClaim.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.persistentVolumeClaim.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.persistentVolumeClaim.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.pod	object	Template for Kafka Pods .
spec.kafka.kafka.template.pod.affinity	object	The pod's affinity rules.

Property	Type	Description
spec.kafka.kafka.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.kafka.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.kafka.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.kafka.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".

Property	Type	Description
spec.kafka.kafka.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafka.template.pod.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
spec.kafka.kafka.template.pod.imagePullSecrets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.kafka.template.pod.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.pod.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.pod.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.pod.priorityClassName	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.
spec.kafka.kafka.template.pod.schedulerName	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.kafka.template.pod.securityContext	object	Configures pod-level security attributes and common container settings.
spec.kafka.kafka.template.pod.securityContext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".

Property	Type	Description
spec.kafka.kafka.template.pod.securityContext.fsGroupChangePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.kafka.template.pod.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.kafka.template.pod.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafka.template.pod.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.kafka.template.pod.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.kafka.template.pod.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafka.template.pod.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafka.template.pod.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.pod.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.kafka.template.pod.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafka.template.pod.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".

Property	Type	Description
spec.kafka.kafka.template.pod.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.pod.securityContext.supplementalGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".
spec.kafka.kafka.template.pod.securityContext.sysctls	array	Sysctls corresponds to the JSON schema field "sysctls".
spec.kafka.kafka.template.pod.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafka.template.pod.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.kafka.template.pod.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafka.template.pod.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.kafka.template.pod.terminationGracePeriodSeconds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.kafka.template.pod.tolerations	array	The pod's tolerations.

Property	Type	Description
spec.kafka.kafka.template.pod.topologySpreadConstraints	array	The pod's topology spread constraints.
spec.kafka.kafka.template.podDisruptionBudget	object	Template for Kafka PodDisruptionBudget .
spec.kafka.kafka.template.podDisruptionBudget.maxUnavailable	integer	Maximum number of unavailable pods to allow automatic Pod eviction. A Pod eviction is allowed when the maxUnavailable number of pods or fewer are unavailable after the eviction. Setting this value to 0 prevents all voluntary evictions, so the pods must be evicted manually. Defaults to 1.
spec.kafka.kafka.template.podDisruptionBudget.metadata	object	Metadata to apply to the PodDisruptionBudgetTemplate resource.
spec.kafka.kafka.template.podDisruptionBudget.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.podDisruptionBudget.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.statefulset	object	Template for Kafka StatefulSet .
spec.kafka.kafka.template.statefulset.metadata	object	Metadata applied to the resource.
spec.kafka.kafka.template.statefulset.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.statefulset.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafka.template.statefulset.podManagementPolicy	string	PodManagementPolicy which will be used for this StatefulSet. Valid values are Parallel and OrderedReady . Defaults to Parallel .
spec.kafka.kafka.template.tlsSidecarContainer	object	Template for the Kafka broker TLS sidecar container.

Property	Type	Description
spec.kafka.kafka.template.tlsSidecarContainer.env	array	Environment variables which should be applied to the container.
spec.kafka.kafka.template.tlsSidecarContainer.securityContext	object	Security context for the container.
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".

Property	Type	Description
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".

Property	Type	Description
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafka.template.tlsSidecarContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.kafka.tlsSidecar	object	TLS sidecar configuration.
spec.kafka.kafka.tlsSidecar.image	string	The docker image for the container.
spec.kafka.kafka.tlsSidecar.livenessProbe	object	Pod liveness checking.
spec.kafka.kafka.tlsSidecar.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.

Property	Type	Description
spec.kafka.kafka.tlsSidecar.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.logLevel	string	The log level for the TLS sidecar. Default value is notice .
spec.kafka.kafka.tlsSidecar.readinessProbe	object	Pod readiness checking.
spec.kafka.kafka.tlsSidecar.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.kafka.tlsSidecar.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafka.tlsSidecar.resources	object	CPU and memory resources to reserve.
spec.kafka.kafka.tlsSidecar.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.kafka.tlsSidecar.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.kafka.tolerations	array	The pod's tolerations.

Property	Type	Description
spec.kafka.kafka.version	string	The kafka broker version. Defaults to {DefaultKafkaVersion}. Consult the user documentation to understand the process required to upgrade or downgrade the version.
spec.kafka.kafkaExporter	object	Configuration of the Kafka Exporter. Kafka Exporter can provide additional metrics, for example lag of consumer group at topic/partition.
spec.kafka.kafkaExporter.enableSaramaLogging	boolean	Enable Sarama logging, a Go client library used by the Kafka Exporter.
spec.kafka.kafkaExporter.groupRegex	string	Regular expression to specify which consumer groups to collect. Default value is <code>.*</code> .
spec.kafka.kafkaExporter.image	string	The docker image for the pods.
spec.kafka.kafkaExporter.livenessProbe	object	Pod liveness check.
spec.kafka.kafkaExporter.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.kafkaExporter.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.kafkaExporter.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafkaExporter.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafkaExporter.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafkaExporter.logging	string	Only log messages with the given severity or above. Valid levels: <code>[debug, info, warn, error, fatal]</code> . Default log level is <code>info</code> .
spec.kafka.kafkaExporter.readinessProbe	object	Pod readiness check.
spec.kafka.kafkaExporter.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
spec.kafka.kafkaExporter.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.kafkaExporter.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.kafkaExporter.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.kafkaExporter.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.kafkaExporter.resources	object	CPU and memory resources to reserve.
spec.kafka.kafkaExporter.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.kafkaExporter.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.kafkaExporter.template	object	Customization of deployment templates and pods.
spec.kafka.kafkaExporter.template.container	object	Template for the Kafka Exporter container.
spec.kafka.kafkaExporter.template.container.env	array	Environment variables which should be applied to the container.
spec.kafka.kafkaExporter.template.container.securityContext	object	Security context for the container.
spec.kafka.kafkaExporter.template.container.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".

Property	Type	Description
spec.kafka.kafkaExporter.template.container.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.kafkaExporter.template.container.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.kafkaExporter.template.container.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.kafkaExporter.template.container.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.kafkaExporter.template.container.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.kafkaExporter.template.container.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.kafkaExporter.template.container.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.kafkaExporter.template.container.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafkaExporter.template.container.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".

Property	Type	Description
spec.kafka.kafkaExporter.template.container.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.kafkaExporter.template.container.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafkaExporter.template.container.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafkaExporter.template.container.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafkaExporter.template.container.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.kafkaExporter.template.container.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafkaExporter.template.container.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.kafkaExporter.template.container.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".

Property	Type	Description
spec.kafka.kafkaExporter.template.container.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafkaExporter.template.container.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.kafkaExporter.template.container.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafkaExporter.template.container.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.kafkaExporter.template.deployment	object	Template for Kafka Exporter Deployment .
spec.kafka.kafkaExporter.template.deployment.metadata	object	Metadata applied to the resource.
spec.kafka.kafkaExporter.template.deployment.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.template.deployment.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.template.pod	object	Template for Kafka Exporter Pods .
spec.kafka.kafkaExporter.template.pod.affinity	object	The pod's affinity rules.

Property	Type	Description
spec.kafka.kafkaExporter.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.kafkaExporter.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.kafkaExporter.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.kafkaExporter.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".

Property	Type	Description
spec.kafka.kafkaExporter.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.kafkaExporter.template.pod.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
spec.kafka.kafkaExporter.template.pod.imagePullSecrets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.kafkaExporter.template.pod.metadata	object	Metadata applied to the resource.
spec.kafka.kafkaExporter.template.pod.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.template.pod.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.template.pod.priorityClassName	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.
spec.kafka.kafkaExporter.template.pod.schedulerName	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.kafkaExporter.template.pod.securityContext	object	Configures pod-level security attributes and common container settings.

Property	Type	Description
spec.kafka.kafkaExporter.template.pod.securityContext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".
spec.kafka.kafkaExporter.template.pod.securityContext.fsGroupChangePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.kafkaExporter.template.pod.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.kafkaExporter.template.pod.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.kafkaExporter.template.pod.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.kafkaExporter.template.pod.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.kafkaExporter.template.pod.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.kafkaExporter.template.pod.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.kafkaExporter.template.pod.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafkaExporter.template.pod.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".

Property	Type	Description
spec.kafka.kafkaExporter.template.pod.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.kafkaExporter.template.pod.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.kafkaExporter.template.pod.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.kafkaExporter.template.pod.securityContext.supplementalGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".
spec.kafka.kafkaExporter.template.pod.securityContext.sysctls	array	Sysctls corresponds to the JSON schema field "sysctls".
spec.kafka.kafkaExporter.template.pod.securityContext.windowOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.kafkaExporter.template.pod.securityContext.windowOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.kafkaExporter.template.pod.securityContext.windowOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.kafkaExporter.template.pod.securityContext.windowOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".

Property	Type	Description
spec.kafka.kafkaExporter.template.pod.terminationGracePeriodSeconds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.kafkaExporter.template.pod.tolerations	array	The pod's tolerations.
spec.kafka.kafkaExporter.template.pod.topologySpreadConstraints	array	The pod's topology spread constraints.
spec.kafka.kafkaExporter.template.service	object	Template for Kafka Exporter Service .
spec.kafka.kafkaExporter.template.service.metadata	object	Metadata applied to the resource.
spec.kafka.kafkaExporter.template.service.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.template.service.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.kafkaExporter.topicRegex	string	Regular expression to specify which topics to collect. Default value is .* .
spec.kafka.maintenanceTimeWindows	array	A list of time windows for maintenance tasks (that is, certificates renewal). Each time window is defined by a cron expression.
spec.kafka.topicOperator	object	Configuration of the Topic Operator.
spec.kafka.topicOperator.affinity	object	Pod affinity rules.
spec.kafka.topicOperator.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".

Property	Type	Description
spec.kafka.topicOperator.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.topicOperator.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.topicOperator.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.topicOperator.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.topicOperator.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.topicOperator.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.topicOperator.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.topicOperator.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".

Property	Type	Description
spec.kafka.topicOperator.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.topicOperator.image	string	The image to use for the Topic Operator.
spec.kafka.topicOperator.jvmOptions	object	JVM Options for pods.
spec.kafka.topicOperator.jvmOptions.-XX	object	A map of -XX options to the JVM.
spec.kafka.topicOperator.jvmOptions.-Xms	string	-Xms option to to the JVM.
spec.kafka.topicOperator.jvmOptions.-Xmx	string	-Xmx option to to the JVM.
spec.kafka.topicOperator.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.topicOperator.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the -D option to the JVM.
spec.kafka.topicOperator.livenessProbe	object	Pod liveness checking.
spec.kafka.topicOperator.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.topicOperator.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.topicOperator.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.topicOperator.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.

Property	Type	Description
spec.kafka.topicOperator.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.topicOperator.logging	object	Logging configuration.
spec.kafka.topicOperator.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.topicOperator.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.topicOperator.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.topicOperator.readinessProbe	object	Pod readiness checking.
spec.kafka.topicOperator.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.topicOperator.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.topicOperator.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.topicOperator.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.topicOperator.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.topicOperator.reconciliationIntervalSeconds	integer	Interval between periodic reconciliations.
spec.kafka.topicOperator.resources	object	CPU and memory resources to reserve.
spec.kafka.topicOperator.resources.limits	object	Limits corresponds to the JSON schema field "limits".

Property	Type	Description
spec.kafka.topicOperator.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.topicOperator.tlsSidecar	object	TLS sidecar configuration.
spec.kafka.topicOperator.tlsSidecar.image	string	The docker image for the container.
spec.kafka.topicOperator.tlsSidecar.livenessProbe	object	Pod liveness checking.
spec.kafka.topicOperator.tlsSidecar.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.topicOperator.tlsSidecar.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.logLevel	string	The log level for the TLS sidecar. Default value is notice .
spec.kafka.topicOperator.tlsSidecar.readinessProbe	object	Pod readiness checking.
spec.kafka.topicOperator.tlsSidecar.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

Property	Type	Description
spec.kafka.topicOperator.tlsSidecar.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.topicOperator.tlsSidecar.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.topicOperator.tlsSidecar.resources	object	CPU and memory resources to reserve.
spec.kafka.topicOperator.tlsSidecar.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.topicOperator.tlsSidecar.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.topicOperator.topicMetadataMaxAttempts	integer	The number of attempts at getting topic metadata.
spec.kafka.topicOperator.watchedNamespace	string	The namespace the Topic Operator should watch.
spec.kafka.topicOperator.zookeeper.sessionTimeoutSeconds	integer	Timeout for the ZooKeeper session.
spec.kafka.zookeeper	object	Configuration of the ZooKeeper cluster.
spec.kafka.zookeeper.affinity	object	The pod's affinity rules.
spec.kafka.zookeeper.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".

Property	Type	Description
spec.kafka.zookeeper.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.zookeeper.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.zookeeper.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.zookeeper.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".

Property	Type	Description
spec.kafka.zookeeper.config	object	The ZooKeeper broker config. Properties with the following prefixes cannot be set: server., dataDir, dataLogDir, clientPort, authProvider, quorum.auth, requireClientAuthScheme, snapshot.trust.empty, standaloneEnabled, reconfigEnabled, 4lw.commands.whitelist, secureClientPort, ssl., serverCnxnFactory, sslQuorum (with the exception of: ssl.protocol, ssl.quorum.protocol, ssl.enabledProtocols, ssl.quorum.enabledProtocols, ssl.ciphersuites, ssl.quorum.ciphersuites, ssl.hostnameVerification, ssl.quorum.hostnameVerification).
spec.kafka.zookeeper.image	string	The docker image for the pods.
spec.kafka.zookeeper.jvmOptions	object	JVM Options for pods.
spec.kafka.zookeeper.jvmOptions.-XX	object	A map of -XX options to the JVM.
spec.kafka.zookeeper.jvmOptions.-Xms	string	-Xms option to to the JVM.
spec.kafka.zookeeper.jvmOptions.-Xmx	string	-Xmx option to to the JVM.
spec.kafka.zookeeper.jvmOptions.gcLoggingEnabled	boolean	Specifies whether the Garbage Collection logging is enabled. The default is false.
spec.kafka.zookeeper.jvmOptions.javaSystemProperties	array	A map of additional system properties which will be passed using the -D option to the JVM.
spec.kafka.zookeeper.livenessProbe	object	Pod liveness checking.
spec.kafka.zookeeper.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.zookeeper.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.zookeeper.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.

Property	Type	Description
spec.kafka.zookeeper.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.zookeeper.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.zookeeper.logging	object	Logging configuration for ZooKeeper.
spec.kafka.zookeeper.logging.loggers	object	A Map from logger name to logger level.
spec.kafka.zookeeper.logging.name	string	The name of the ConfigMap from which to get the logging configuration.
spec.kafka.zookeeper.logging.type	string	Logging type, must be either 'inline' or 'external'.
spec.kafka.zookeeper.metrics	object	The Prometheus JMX Exporter configuration. See https://github.com/prometheus/jmx_exporter for details of the structure of this configuration.
spec.kafka.zookeeper.metricsConfig	object	Metrics configuration.
spec.kafka.zookeeper.metricsConfig.type	string	Metrics type. Only 'jmxPrometheusExporter' supported currently.
spec.kafka.zookeeper.metricsConfig.valueFrom	object	ConfigMap where the Prometheus JMX Exporter configuration is stored. For details of the structure of this configuration, see the {JMXExporter}.
spec.kafka.zookeeper.metricsConfig.valueFrom.configMapKeyRef	object	Reference to the key in the ConfigMap containing the metrics configuration.
spec.kafka.zookeeper.metricsConfig.valueFrom.configMapKeyRef.key	string	Key corresponds to the JSON schema field "key".
spec.kafka.zookeeper.metricsConfig.valueFrom.configMapKeyRef.name	string	Name corresponds to the JSON schema field "name".

Property	Type	Description
spec.kafka.zookeeper.metricsConfig.valueFrom.configMapKeyRef.optional	boolean	Optional corresponds to the JSON schema field "optional".
spec.kafka.zookeeper.readinessProbe	object	Pod readiness checking.
spec.kafka.zookeeper.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.zookeeper.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.zookeeper.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.zookeeper.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.zookeeper.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.zookeeper.replicas	integer	The number of pods in the cluster.
spec.kafka.zookeeper.resources	object	CPU and memory resources to reserve.
spec.kafka.zookeeper.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.zookeeper.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.zookeeper.storage	object	Storage configuration (disk). Cannot be updated.
spec.kafka.zookeeper.storage.class	string	The storage class to use for dynamic volume allocation.
spec.kafka.zookeeper.storage.deleteClaim	boolean	Specifies if the persistent volume claim has to be deleted when the cluster is un-deployed.
spec.kafka.zookeeper.storage.id	integer	Storage identification number. It is mandatory only for storage volumes defined in a storage of type 'jbod'.

Property	Type	Description
spec.kafka.zookeeper.storage.overrides	array	Overrides for individual brokers. The overrides field allows to specify a different configuration for different brokers.
spec.kafka.zookeeper.storage.selector	object	Specifies a specific persistent volume to use. It contains key:value pairs representing labels for selecting such a volume.
spec.kafka.zookeeper.storage.size	string	When type=persistent-claim, defines the size of the persistent volume claim (i.e 1Gi). Mandatory when type=persistent-claim.
spec.kafka.zookeeper.storage.sizeLimit	string	When type=ephemeral, defines the total amount of local storage required for this EmptyDir volume (for example 1Gi).
spec.kafka.zookeeper.storage.type	string	Storage type, must be either 'ephemeral' or 'persistent-claim'.
spec.kafka.zookeeper.template	object	Template for ZooKeeper cluster resources. The template allows users to specify how are the StatefulSet , Pods and Services generated.
spec.kafka.zookeeper.template.clientService	object	Template for ZooKeeper client Service .
spec.kafka.zookeeper.template.clientService.metadata	object	Metadata applied to the resource.
spec.kafka.zookeeper.template.clientService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.clientService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.nodesService	object	Template for ZooKeeper nodes Service .
spec.kafka.zookeeper.template.nodesService.metadata	object	Metadata applied to the resource.
spec.kafka.zookeeper.template.nodesService.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.nodesService.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .

Property	Type	Description
spec.kafka.zookeeper.template.persistentVolumeClaim	object	Template for all ZooKeeper PersistentVolumeClaims .
spec.kafka.zookeeper.template.persistentVolumeClaim.metadata	object	Metadata applied to the resource.
spec.kafka.zookeeper.template.persistentVolumeClaim.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.persistentVolumeClaim.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.pod	object	Template for ZooKeeper Pods .
spec.kafka.zookeeper.template.pod.affinity	object	The pod's affinity rules.
spec.kafka.zookeeper.template.pod.affinity.nodeAffinity	object	NodeAffinity corresponds to the JSON schema field "nodeAffinity".
spec.kafka.zookeeper.template.pod.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	object	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".

Property	Type	Description
spec.kafka.zookeeper.template.pod.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerms	array	NodeSelectorTerms corresponds to the JSON schema field "nodeSelectorTerms".
spec.kafka.zookeeper.template.pod.affinity.podAffinity	object	PodAffinity corresponds to the JSON schema field "podAffinity".
spec.kafka.zookeeper.template.pod.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.template.pod.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.template.pod.affinity.podAntiAffinity	object	PodAntiAffinity corresponds to the JSON schema field "podAntiAffinity".
spec.kafka.zookeeper.template.pod.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	array	PreferredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "preferredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.template.pod.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	array	RequiredDuringSchedulingIgnoredDuringExecution corresponds to the JSON schema field "requiredDuringSchedulingIgnoredDuringExecution".
spec.kafka.zookeeper.template.pod.hostAliases	array	The pod's HostAliases. HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Property	Type	Description
spec.kafka.zookeeper.template.pod.imagePullSecrets	array	List of references to secrets in the same namespace to use for pulling any of the images used by this Pod. When the STRIMZI_IMAGE_PULL_SECRETS environment variable in Cluster Operator and the imagePullSecrets option are specified, only the imagePullSecrets variable is used and the STRIMZI_IMAGE_PULL_SECRETS variable is ignored.
spec.kafka.zookeeper.template.pod.metadata	object	Metadata applied to the resource.
spec.kafka.zookeeper.template.pod.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.pod.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.pod.priorityClassName	string	The name of the priority class used to assign priority to the pods. For more information about priority classes, see {K8sPriorityClass}.
spec.kafka.zookeeper.template.pod.schedulerName	string	The name of the scheduler used to dispatch this Pod . If not specified, the default scheduler will be used.
spec.kafka.zookeeper.template.pod.securityContext	object	Configures pod-level security attributes and common container settings.
spec.kafka.zookeeper.template.pod.securityContext.fsGroup	integer	FsGroup corresponds to the JSON schema field "fsGroup".
spec.kafka.zookeeper.template.pod.securityContext.fsGroupChangePolicy	string	FsGroupChangePolicy corresponds to the JSON schema field "fsGroupChangePolicy".
spec.kafka.zookeeper.template.pod.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".
spec.kafka.zookeeper.template.pod.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".

Property	Type	Description
spec.kafka.zookeeper.template.pod.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.zookeeper.template.pod.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.zookeeper.template.pod.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.zookeeper.template.pod.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.zookeeper.template.pod.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.pod.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.zookeeper.template.pod.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".
spec.kafka.zookeeper.template.pod.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.zookeeper.template.pod.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.pod.securityContext.supplementalGroups	array	SupplementalGroups corresponds to the JSON schema field "supplementalGroups".

Property	Type	Description
spec.kafka.zookeeper.template.pod.securityContext.sysctl	array	Sysctl corresponds to the JSON schema field "sysctl".
spec.kafka.zookeeper.template.pod.securityContext.windowOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.zookeeper.template.pod.securityContext.windowOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.zookeeper.template.pod.securityContext.windowOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.zookeeper.template.pod.securityContext.windowOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.zookeeper.template.pod.terminationGracePeriodSeconds	integer	The grace period is the duration in seconds after the processes running in the pod are sent a termination signal, and the time when the processes are forcibly halted with a kill signal. Set this value to longer than the expected cleanup time for your process. Value must be a non-negative integer. A zero value indicates delete immediately. You might need to increase the grace period for very large Kafka clusters, so that the Kafka brokers have enough time to transfer their work to another broker before they are terminated. Defaults to 30 seconds.
spec.kafka.zookeeper.template.pod.tolerations	array	The pod's tolerations.
spec.kafka.zookeeper.template.pod.topologySpreadConstraints	array	The pod's topology spread constraints.
spec.kafka.zookeeper.template.podDisruptionBudget	object	Template for ZooKeeper PodDisruptionBudget .

Property	Type	Description
spec.kafka.zookeeper.template.podDisruptionBudget.maxUnavailable	integer	Maximum number of unavailable pods to allow automatic Pod eviction. A Pod eviction is allowed when the maxUnavailable number of pods or fewer are unavailable after the eviction. Setting this value to 0 prevents all voluntary evictions, so the pods must be evicted manually. Defaults to 1.
spec.kafka.zookeeper.template.podDisruptionBudget.metadata	object	Metadata to apply to the PodDisruptionBudgetTemplate resource.
spec.kafka.zookeeper.template.podDisruptionBudget.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.podDisruptionBudget.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.statefulset	object	Template for ZooKeeper StatefulSet .
spec.kafka.zookeeper.template.statefulset.metadata	object	Metadata applied to the resource.
spec.kafka.zookeeper.template.statefulset.metadata.annotations	object	Annotations added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.statefulset.metadata.labels	object	Labels added to the resource template. Can be applied to different resources such as StatefulSets , Deployments , Pods , and Services .
spec.kafka.zookeeper.template.statefulset.podManagementPolicy	string	PodManagementPolicy which will be used for this StatefulSet. Valid values are Parallel and OrderedReady . Defaults to Parallel .
spec.kafka.zookeeper.template.tlsSidecarContainer	object	Template for the Zookeeper server TLS sidecar container. The TLS sidecar is not used anymore and this option will be ignored.
spec.kafka.zookeeper.template.tlsSidecarContainer.env	array	Environment variables which should be applied to the container.

Property	Type	Description
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext	object	Security context for the container.
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".

Property	Type	Description
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".

Property	Type	Description
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.windowsOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.windowsOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.zookeeper.template.tlsSidecarContainer.securityContext.windowsOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.zookeeper.template.zookeeperContainer	object	Template for the ZooKeeper container.
spec.kafka.zookeeper.template.zookeeperContainer.env	array	Environment variables which should be applied to the container.

Property	Type	Description
spec.kafka.zookeeper.template.zookeeperContainer.securityContext	object	Security context for the container.
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation corresponds to the JSON schema field "allowPrivilegeEscalation".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.capabilities	object	Capabilities corresponds to the JSON schema field "capabilities".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.capabilities.add	array	Add corresponds to the JSON schema field "add".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.capabilities.drop	array	Drop corresponds to the JSON schema field "drop".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.privileged	boolean	Privileged corresponds to the JSON schema field "privileged".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.procMount	string	ProcMount corresponds to the JSON schema field "procMount".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.readOnlyRootFilesystem	boolean	ReadOnlyRootFilesystem corresponds to the JSON schema field "readOnlyRootFilesystem".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.runAsGroup	integer	RunAsGroup corresponds to the JSON schema field "runAsGroup".

Property	Type	Description
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.runAsNonRoot	boolean	RunAsNonRoot corresponds to the JSON schema field "runAsNonRoot".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.runAsUser	integer	RunAsUser corresponds to the JSON schema field "runAsUser".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seLinuxOptions	object	SeLinuxOptions corresponds to the JSON schema field "seLinuxOptions".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seLinuxOptions.level	string	Level corresponds to the JSON schema field "level".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seLinuxOptions.role	string	Role corresponds to the JSON schema field "role".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seLinuxOptions.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seLinuxOptions.user	string	User corresponds to the JSON schema field "user".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seccompProfile	object	SeccompProfile corresponds to the JSON schema field "seccompProfile".

Property	Type	Description
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seccompProfile.localhostProfile	string	LocalhostProfile corresponds to the JSON schema field "localhostProfile".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.seccompProfile.type	string	Type corresponds to the JSON schema field "type".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.windowOptions	object	WindowsOptions corresponds to the JSON schema field "windowsOptions".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.windowOptions.gmsaCredentialSpec	string	GmsaCredentialSpec corresponds to the JSON schema field "gmsaCredentialSpec".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.windowOptions.gmsaCredentialSpecName	string	GmsaCredentialSpecName corresponds to the JSON schema field "gmsaCredentialSpecName".
spec.kafka.zookeeper.template.zookeeperContainer.securityContext.windowOptions.runAsUserName	string	RunAsUserName corresponds to the JSON schema field "runAsUserName".
spec.kafka.zookeeper.tlsSidecar	object	TLS sidecar configuration. The TLS sidecar is not used anymore and this option will be ignored.
spec.kafka.zookeeper.tlsSidecar.image	string	The docker image for the container.
spec.kafka.zookeeper.tlsSidecar.livenessProbe	object	Pod liveness checking.

Property	Type	Description
spec.kafka.zookeeper.tlsSidecar.livenessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.livenessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.zookeeper.tlsSidecar.livenessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.livenessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.livenessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.logLevel	string	The log level for the TLS sidecar. Default value is notice .
spec.kafka.zookeeper.tlsSidecar.readinessProbe	object	Pod readiness checking.
spec.kafka.zookeeper.tlsSidecar.readinessProbe.failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.readinessProbe.initialDelaySeconds	integer	The initial delay before first the health is first checked. Default to 15 seconds. Minimum value is 0.
spec.kafka.zookeeper.tlsSidecar.readinessProbe.periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.readinessProbe.successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness. Minimum value is 1.
spec.kafka.zookeeper.tlsSidecar.readinessProbe.timeoutSeconds	integer	The timeout for each attempted health check. Default to 5 seconds. Minimum value is 1.

Property	Type	Description
spec.kafka.zookeeper.tlsSidecar.resources	object	CPU and memory resources to reserve.
spec.kafka.zookeeper.tlsSidecar.resources.limits	object	Limits corresponds to the JSON schema field "limits".
spec.kafka.zookeeper.tlsSidecar.resources.requests	object	Requests corresponds to the JSON schema field "requests".
spec.kafka.zookeeper.tolerations	array	The pod's tolerations.
spec.kafkaAccessSecret	string	KafkaAccessSecret when set will configure the application to use the provided kafka credentials instead of deploying its own.
spec.license	object	LicenseAccept is used to accept the license agreement
spec.license.accept	boolean	(No Description)
spec.postgres	object	Postgres property
spec.postgres.accessSecret	string	Access secret name that contains the Postgres service name and port, username, password, and CA certificate.
spec.size	string	(No Description)
spec.storageClass	string	StorageClass is the storage class used for PVCs. If not set it will fall back to the cluster default when available.

status

Description: CommonStatus that can be used with StatusUpdater

Type: object

Table 1. Description of AIOpsEdge/v1beta1 status

Property	Type	Description
status	object	CommonStatus that can be used with StatusUpdater
status.conditions	array	(No Description)
status.observedGeneration	integer	(No Description)
status.phase	string	(No Description)
status.resources	object	Resources contained in status

AIOpsUI/v1

For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#). AIOpsUI is the Schema for the aiopsuis API.

spec

Description: (No Description)

Type: object

Table 1. Description of AIOpsUI/v1 spec

Property	Type	Description
spec	object	(No Description)
spec.configuration	object	Additional configuration for the UI deployment
spec.configuration.topologyModel	string	TopologyModel - Specify the terminology to use in CP4WAIOps when defining collections of topology resource groups. Valid values are "application" or "service". If this setting is not specified, the default value will be "application".
spec.container	object	ContainerSpec - possible configurations around containers
spec.container.applicationUiApi	object	(No Description)
spec.container.applicationUiApi.container	object	ContainerSpec - possible configurations around containers
spec.container.applicationUiApi.container.containerPort	integer	(No Description)
spec.container.applicationUiApi.container.env	array	(No Description)
spec.container.applicationUiApi.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.applicationUiApi.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.applicationUiApi.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.applicationUiApi.container.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.applicationUiApi.image	object	ImageSpec - specification around how to pull an image.
spec.container.applicationUiApi.image.path	string	(No Description)

Property	Type	Description
spec.container.applicationUiApi.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.applicationUiApi.image.pullSecret	string	(No Description)
spec.container.applicationUiApi.tlsSecretName	string	(No Description)
spec.container.containerPort	integer	(No Description)
spec.container.env	array	(No Description)
spec.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.uiApi	object	(No Description)
spec.container.uiApi.container	object	ContainerSpec - possible configurations around containers
spec.container.uiApi.container.containerPort	integer	(No Description)
spec.container.uiApi.container.env	array	(No Description)
spec.container.uiApi.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.uiApi.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.uiApi.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
spec.container.u iApi.container.re sources.request s	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iApi.image	object	ImageSpec - specification around how to pull an image.
spec.container.u iApi.image.path	string	(No Description)
spec.container.u iApi.image.pullP olicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.u iApi.image.pullS ecret	string	(No Description)
spec.container.u iApi.tlsSecretNa me	string	(No Description)
spec.container.u iBundleApi	object	(No Description)
spec.container.u iBundleApi.cont ainer	object	ContainerSpec - possible configurations around containers
spec.container.u iBundleApi.cont ainer.containerP ort	integer	(No Description)
spec.container.u iBundleApi.cont ainer.env	array	(No Description)
spec.container.u iBundleApi.cont ainer.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.u iBundleApi.cont ainer.resources. claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.u iBundleApi.cont ainer.resources.l imits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iBundleApi.cont ainer.resources. requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

Property	Type	Description
spec.container.u iBundleApi.imag e	object	ImageSpec - specification around how to pull an image.
spec.container.u iBundleApi.imag e.path	string	(No Description)
spec.container.u iBundleApi.imag e.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.u iBundleApi.imag e.pullSecret	string	(No Description)
spec.container.u iBundleApi.tlsSe cretName	string	(No Description)
spec.container.u iContent	object	(No Description)
spec.container.u iContent.contain er	object	ContainerSpec - possible configurations around containers
spec.container.u iContent.contain er.containerPort	integer	(No Description)
spec.container.u iContent.contain er.env	array	(No Description)
spec.container.u iContent.contain er.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.u iContent.contain er.resources.clai ms	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.u iContent.contain er.resources.limi ts	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iContent.contain er.resources.req uests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iContent.image	object	ImageSpec - specification around how to pull an image.
spec.container.u iContent.image. path	string	(No Description)

Property	Type	Description
spec.container.u iContent.image. pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.u iContent.image. pullSecret	string	(No Description)
spec.container.u iContent.tlsSecr etName	string	(No Description)
spec.container.u iServer	object	(No Description)
spec.container.u iServer.container r	object	ContainerSpec - possible configurations around containers
spec.container.u iServer.container .containerPort	integer	(No Description)
spec.container.u iServer.container .env	array	(No Description)
spec.container.u iServer.container .resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.u iServer.container .resources.claim s	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.u iServer.container .resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iServer.container .resources.requ ests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iServer.image	object	ImageSpec - specification around how to pull an image.
spec.container.u iServer.image.pa th	string	(No Description)
spec.container.u iServer.image.pu llPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.u iServer.image.pu llSecret	string	(No Description)

Property	Type	Description
spec.container.uiServer.tlsSecretName	string	(No Description)
spec.elasticSearch	object	Contact details for ElasticSearch
spec.elasticSearch.bindingSecret	string	Data store binding secret
spec.image	object	ImageSpec - specification around how to pull an image.
spec.image.path	string	(No Description)
spec.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.image.pullSecret	string	(No Description)
spec.license	object	License - license spec determines whether a license has been accepted
spec.license.accept	boolean	License acceptance switch
spec.name	string	Name of the service being deployed
spec.redis	object	Name of the service being deployed
spec.redis.bindingSecret	string	Data store binding secret
spec.replicas	integer	(No Description)
spec.size	string	(No Description)
spec.tlsSecretName	string	(No Description)
spec.version	string	Version of the csv deploying
spec.zenserviceName	string	name of the zenservice cr that is used for this deployment

status

Description: (No Description)

Type: object

Table 1. Description of AIOpsUI/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

AIOpsUIExtension/v1

spec

Description: AIOpsUIExtensionSpec defines the desired state of AIOpsUIExtension

Type: object

Table 1. Description of AIOpsUIExtension/v1 spec

Property	Type	Description
spec	object	AIOpsUIExtensionSpec defines the desired state of AIOpsUIExtension
spec.frameSrcUrls	string	FrameSourceUrls contains any frame-srcs that will be applied to our Akora UI server CSP.
spec.menuRoutes	(No Type)	MenuRoutes contains menu configurations, that will be mapped to the formats required by Zen and Akora.
spec.routes	(No Type)	Routes contains page configuration and bundle configurations.

status

Description: (No Description)

Type: object

Table 1. Description of AIOpsUIExtension/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

AIManager/v1beta1

AIManager Deploys the IBM AIOps AI Manager service components of IBM Cloud Pak for AIOps. For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#).

spec

Description: AIManagerSpec defines the desired state of AIManager

Type: object

Table 1. Description of AIManager/v1beta1 spec

Property	Type	Description

Property	Type	Description
spec	object	AIManagerSpec defines the desired state of AIManager
spec.aimanagerValues	object	Plumbing Values Override (hidden)
spec.cartridge	object	Cartridge Name
spec.cartridge.name	string	Cartridge Name
spec.certificates	object	Certificates contains information required to create and trust certificates for internal TLS communication.
spec.certificates.caConfigMapKey	string	Key in the ConfigMap identified by caConfigMapName that contains root certificates to be trusted for internal TLS.
spec.certificates.caConfigMapName	string	Name of the ConfigMap that contains root certificates to be trusted for internal TLS.
spec.certificates.issuer	string	Name of the cert-manager.io/Issuer that will be used to create TLS certificates.
spec.chatOpsDisabled	boolean	ChatOpsDisabled (hidden)
spec.chatopsAuthentication	object	ChatopsAuthentication (hidden)
spec.chatopsAuthentication.configmapName	string	(No Description)
spec.chatopsAuthentication.secretName	string	(No Description)
spec.cronjobConfig	object	Cronjob Configuration Properties
spec.cronjobConfig.unitCount	integer	Unit Count
spec.debug	object	Debug Options (hidden)
spec.debug.devPullSecrets	array	(No Description)
spec.debug.disableAIManager	boolean	(No Description)
spec.debug.disableElastic	boolean	(No Description)
spec.debug.disableKafka	boolean	(No Description)
spec.debug.disableModeltrain	boolean	Note: this element is no longer used and should be removed in the next major version of this API
spec.debug.disableZen	boolean	(No Description)
spec.debug.enableCustomSCC	boolean	(No Description)
spec.debug.useDockerIOForKafka	boolean	(No Description)
spec.directFlink	boolean	(No Description)
spec.elasticsearch	object	Elastic Search
spec.elasticsearch.elasticSuperuserSecretName	string	ElasticUserSecretName
spec.elasticsearch.elasticUserSecretName	string	ElasticUserSecretName
spec.eventProcessor	object	Event Processor

Property	Type	Description
spec.eventProcessor.name	string	Event Processor Name
spec.existingPullSecret	object	Existing Pull Secret
spec.existingPullSecret.name	string	Secret Name
spec.flink	object	Flink
spec.flink.fdSecretName	string	Flink Deployment Secret Name
spec.flink.secretName	string	Flink Secret Name
spec.integratorProxy	object	Integrator Proxy Config Properties
spec.integratorProxy.enabled	boolean	A proxy outgoing ChatOps routes through.
spec.integratorProxy.proxyRoute	string	(No Description)
spec.kafka	object	Kafka
spec.kafka.secretName	string	Kafka Secret Name
spec.license	object	Please accept the licenses to allow for installation
spec.license.accept	boolean	I have read and accept the license: https://ibm.biz/cp4aiops-470-license
spec.logAnomaly	object	LogAnomalyDetectorConfig (hidden)
spec.logAnomaly.limits	object	(No Description)
spec.logAnomaly.limits.cpu	string	(No Description)
spec.logAnomaly.limits.memory	string	(No Description)
spec.logAnomaly.replicas	integer	(No Description)
spec.logAnomaly.requests	object	(No Description)
spec.logAnomaly.requests.cpu	string	(No Description)
spec.logAnomaly.requests.memory	string	(No Description)
spec.logAnomaly.serverPools	integer	(No Description)
spec.logAnomalyDisabled	boolean	LogAnomalyDisabled (hidden)
spec.minio	object	Minio Properties
spec.minio.customSizing	object	MinioCustomResources (hidden)
spec.minio.customSizing.limits	object	(No Description)
spec.minio.customSizing.limits.cpu	string	(No Description)

Property	Type	Description
spec.minio.customSizing.limits.memory	string	(No Description)
spec.minio.customSizing.replicas	integer	(No Description)
spec.minio.customSizing.requests	object	(No Description)
spec.minio.customSizing.requests.cpu	string	(No Description)
spec.minio.customSizing.requests.memory	string	(No Description)
spec.minio.customSizing.serverPools	integer	(No Description)
spec.minio.storage	object	The Minio storage options
spec.minio.storage.storageClassName	string	The StorageClassName
spec.minio.storage.storageSize	string	The Size of Storage (format example "100Gi")
spec.modelTrain	object	Model Train Properties
spec.modelTrain.maxLearnings	integer	Maximum number of training to run at once
spec.modelTrain.pollInterval	integer	Poll Interval for the Queue
spec.modelTrain.sizeLimit	integer	Size Limit for the Queue
spec.postgresDB	object	Postgres Database Instance Properties
spec.postgresDB.edbSecretName	string	EDB Postgresql Database Connection Secret Name
spec.postgresDB.secretName	string	Postgresql Database Connection Secret Name
spec.redis	object	Redis
spec.redis.bindingSecretName	string	Redis Binding Secret Name
spec.repo	string	The private registry to pull images from instead of cp.icr.io. Leave blank if a private registry is not used. All images must be pushed to repo via mirroring.
spec.size	string	Preconfigured HA Sizing Options
spec.storage	object	The storage options
spec.storage.enabled	boolean	Enable Storage
spec.storage.storageClassName	string	The StorageClassName
spec.ticketAnalysisDisabled	boolean	TicketAnalysisDisabled (hidden)
spec.version	string	Version
spec.zen	object	The Zen options
spec.zen.deploy	boolean	Flag to deploy our own cpd in the same namespace as aimanager
spec.zen.ignoreReady	boolean	Flag to deploy zen customization even if not in ready state
spec.zen.instanceId	string	ID of Zen Service Instance

Property	Type	Description
spec.zen.serviceInstanceName	string	Name of the ZenService Instance object
spec.zen.serviceNamespace	string	Namespace of the ZenService Instance
spec.zen.storage	object	The Zen storage options
spec.zen.storage.storageClassName	string	The StorageClassName

status

Description: AIManagerStatus defines the observed state of AIManager

Type: object

Table 1. Description of AIManager/v1beta1 status

Property	Type	Description
status	object	AIManagerStatus defines the observed state of AIManager
status.message	string	Status information from AIManager
status.phase	string	Current phase of AIManager.
status.versions	object	Information on versions for AI Manager
status.versions.available	object	Available properties
status.versions.available.versions	array	Possible available versions for AI Manager
status.versions.reconciled	string	Current version of the reconciled instance

Algorithm/v1alpha1

Algorithm is the Schema for the algorithms API

spec

Description: AlgorithmSpec defines the desired state of Algorithm

Type: object

Table 1. Description of Algorithm/v1alpha1 spec

Property	Type	Description
spec	object	AlgorithmSpec defines the desired state of Algorithm
spec.description	string	(No Description)
spec.enabled	boolean	(No Description)
spec.manifestBase64	string	(No Description)
spec.name	string	(No Description)
spec.registered	boolean	(No Description)
spec.runtime	string	(No Description)

Property	Type	Description
spec.usedStatus	array	(No Description)
spec.version	string	(No Description)

status

Description: AlgorithmStatus defines the observed state of Algorithm

Type: object

Table 1. Description of Algorithm/v1alpha1 status

Property	Type	Description
status	object	AlgorithmStatus defines the observed state of Algorithm

Algorithm/v1beta1

Algorithm is the Schema for the algorithms API

spec

Description: AlgorithmSpec defines the desired state of Algorithm

Type: object

Table 1. Description of Algorithm/v1beta1 spec

Property	Type	Description
spec	object	AlgorithmSpec defines the desired state of Algorithm
spec.description	string	(No Description)
spec.enabled	boolean	(No Description)
spec.manifestBase64	string	(No Description)
spec.name	string	(No Description)
spec.registered	boolean	(No Description)
spec.runtime	string	(No Description)
spec.usedStatus	array	(No Description)
spec.version	string	(No Description)

status

Description: AlgorithmStatus defines the observed state of Algorithm

Type: object

Table 1. Description of Algorithm/v1beta1 status

Property	Type	Description
status	object	AlgorithmStatus defines the observed state of Algorithm

ASM/v1beta1

spec

Description: ASMSpec defines the desired state of ASM

Type: object

Table 1. Description of ASM/v1beta1 spec

Property	Type	Description
spec	object	ASMSpec defines the desired state of ASM
spec.advanced	object	Advanced image properties
spec.advanced.imagePullPolicy	string	The default pull policy is IfNotPresent, which causes the Kubelet to skip pulling an image that already exists.
spec.advanced.imagePullRepository	string	Docker registry that all component images are pulled from
spec.asm	object	Topology components
spec.asm.aaionap	boolean	Do you want to deploy AAI ONAP Service
spec.asm.almObserver	boolean	Do you want to deploy ALM Observer
spec.asm.ansibleawxObserver	boolean	Do you want to deploy Ansible AWX Observer
spec.asm.appDisco	object	Application Discovery properties
spec.asm.appDisco.caCertProperty	string	Cert Property
spec.asm.appDisco.caCertSecret	string	Cert Secret
spec.asm.appDisco.caKeyProperty	string	Key Property
spec.asm.appDisco.certSecret	string	DB2 Certificate Secret
spec.asm.appDisco.dbsecret	string	DB2 secret
spec.asm.appDisco.dburl	string	DB2 host to connect
spec.asm.appDisco.enabled	boolean	Enable Application Discovery and Application Discovery Observer
spec.asm.appDisco.scaleDS	integer	(No Description)
spec.asm.appDisco.scaleSSS	integer	(No Description)
spec.asm.appDisco.secure	boolean	Secure DB connection
spec.asm.appDisco.tlsSecret	string	(No Description)
spec.asm.appdynamicsObserver	boolean	Do you want to deploy AppDynamics Observer
spec.asm.awsObserver	boolean	Do you want to deploy AWS Observer

Property	Type	Description
spec.asm.azureObserver	boolean	Do you want to deploy Microsoft Azure Observer
spec.asm.bigcloudfabricObserver	boolean	Do you want to deploy BigCloud Fabric Observer
spec.asm.bigfixinventoryObserver	boolean	Do you want to deploy BigFix Inventory Observer
spec.asm.cienablueplanetObserver	boolean	Do you want to deploy Ciena Blue Planet Observer
spec.asm.ciscoaciObserver	boolean	Do you want to deploy Cisco ACI Observer
spec.asm.contrailObserver	boolean	Do you want to deploy Juniper Contrail Observer
spec.asm.datadogObserver	boolean	Do you want to deploy Datadog Observer
spec.asm.dnsObserver	boolean	Do you want to deploy DNS Observer
spec.asm.dockerObserver	boolean	Do you want to deploy Docker Observer
spec.asm.dynatraceObserver	boolean	Do you want to deploy Dynatrace Observer
spec.asm.fileObserver	boolean	Do you want to deploy File Observer
spec.asm.gitlabObserver	boolean	Do you want to deploy GitLab Observer
spec.asm.googlecloudObserver	boolean	Do you want to deploy Google Cloud Platform Observer
spec.asm.hpnmfvdObserver	boolean	Do you want to deploy HPNFMVD Observer
spec.asm.iafCartridgeRequirementsName	string	IAF CartridgeRequirements Instance Name for connection to IAF ElasticSearch
spec.asm.iafElasticBindingSecret	string	IAF ElasticSearch Binding Secret for connection to IAF ElasticSearch
spec.asm.iafKafkaBindingSecret	string	IAF Kafka Binding Secret for connection to IAF Kafka
spec.asm.ibmcloudObserver	boolean	Do you want to deploy IBM Cloud Observer (default = false)
spec.asm.itnmObserver	boolean	Do you want to deploy ITNM Observer
spec.asm.jenkinsObserver	boolean	Do you want to deploy Jenkins Observer
spec.asm.junipercsoObserver	boolean	Do you want to deploy Juniper Networks CSO Observer
spec.asm.kubernetesObserver	boolean	Do you want to deploy Kubernetes Observer (default = true)?
spec.asm.netDisco	boolean	Do you want to deploy Network Discovery and Network Discovery Observer
spec.asm.newrelicObserver	boolean	Do you want to deploy New Relic Observer
spec.asm.openstackObserver	boolean	Do you want to deploy OpenStack Observer
spec.asm.rancherObserver	boolean	Do you want to deploy Rancher Observer

Property	Type	Description
spec.asm.restObserver	boolean	Do you want to deploy REST Observer
spec.asm.sdconapObserver	boolean	Do you want to deploy SDC ONAP Observer
spec.asm.servicenowObserver	boolean	Do you want to deploy ServiceNow Observer
spec.asm.sevoneObserver	boolean	Do you want to deploy SevOne Observer
spec.asm.taddmObserver	boolean	Do you want to deploy TADDM Observer
spec.asm.userServiceUrl	string	CEM User Service Url connection
spec.asm.viptelaObserver	boolean	Do you want to deploy Viptela Observer
spec.asm.vmvcenterObserver	boolean	Do you want to deploy VMware VCenter Observer
spec.asm.vmwarensxObserver	boolean	Do you want to deploy VMware NSX Observer
spec.asm.zabbixObserver	boolean	Do you want to deploy Zabbix Observer
spec.clusterDomain	string	Use the fully qualified domain name (FQDN) to specify the cluster domain: apps. ...com
spec.deploymentType	string	Deployment type (small or large)
spec.entitlementSecret	string	Entitlement secret to pull images
spec.helmValuesASM	object	(Not shown within UI)
spec.ingress	object	Vanilla Kube (Not shown within UI)
spec.ingress.port	string	(No Description)
spec.ingress.prefixWithReleaseName	boolean	(No Description)
spec.ingress.tlsSecret	string	(No Description)
spec.license	object	License properties
spec.license.accept	boolean	If you have purchased IBM Cloud Pak for AIOps, you must read and accept this license: https://ibm.biz/cp4aiops-470-license .
spec.persistence	object	Persistence properties
spec.persistence.enabled	boolean	Enable persistence storage
spec.persistence.storageClass	string	Storage Class
spec.persistence.storageSizeCassandraBackup	string	Storage Size Cassandra Backup
spec.persistence.storageSizeCassandraData	string	Storage Size Cassandra Data
spec.persistence.storageSizeElastic	string	Storage Size Elasticsearch
spec.persistence.storageSizeFileObserver	string	Storage size File Observer .
spec.persistence.storageSizeKafka	string	Storage Size Kafka
spec.persistence.storageSizeSdconapObserver	string	Storage Size SDCONAP Observer

Property	Type	Description
spec.persistence.storageSizeZookeeper	string	Storage Size Zookeeper
spec.version	string	Version
spec.zen	object	The Zen Parameters
spec.zen.deploy	boolean	Flag to deploy our own cpd in the same namespace as aimanager
spec.zen.ignoreReady	boolean	Flag to deploy zen customization even if not in ready state
spec.zen.instanceId	string	ID of Zen Service Instance
spec.zen.serviceInstanceName	string	Name of the ZenService Instance object
spec.zen.serviceNamespace	string	Namespace of the ZenService Instance
spec.zen.storage	object	The Zen storage options
spec.zen.storage.storageClassName	string	The StorageClassName

status

Description: (No Description)

Type: object

Table 1. Description of ASM/v1beta1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	Resources managed by Netcool Agile Service Manager
status.message	string	(No Description)
status.phase	string	Current phase of installed full Cloud deployment.
status.versions	object	Information on versions for Agile Service Manager
status.versions.available	object	(No Description)
status.versions.available.versions	array	Possible available versions for Agile Service Manager
status.versions.reconciled	string	Current version of the reconciled instance

BaseUI/v1

For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#).

spec

Description: (No Description)

Type: object

Table 1. Description of BaseUI/v1 spec

Property	Type	Description
spec	object	(No Description)
spec.bedrockVersion	string	(No Description)
spec.certificates	object	Certificates contains information required to create and trust certificates for internal TLS communication.
spec.certificates.caConfigMapKey	string	Key in the ConfigMap identified by caConfigMapName that contains root certificates to be trusted for internal TLS.
spec.certificates.caConfigMapName	string	Name of the ConfigMap that contains root certificates to be trusted for internal TLS.
spec.certificates.issuer	string	Name of the cert-manager.io/Issuer that will be used to create TLS certificates.
spec.configuration	object	Additional configuration for the UI deployment
spec.configuration.topologyModel	string	TopologyModel - Specify the terminology to use in CP4WAI Ops when defining collections of topology resource groups. Valid values are "application" or "service". If this setting is not specified, the default value will be "application".
spec.container	object	ContainerSpec - possible configurations around containers
spec.container.containerPort	integer	(No Description)
spec.container.env	array	(No Description)
spec.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.image	object	ImageSpec - specification around how to pull an image.
spec.image.path	string	(No Description)
spec.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.image.pullSecret	string	(No Description)
spec.license	object	License - license spec determines whether a license has been accepted

Property	Type	Description
spec.license.accept	boolean	License acceptance switch
spec.modules	object	Modules is a list of module settings for the operator. This value will be default for 4.6.x and will eventually be required.
spec.modules.groupProfiles	object	(No Description)
spec.modules.groupProfiles.enabled	boolean	Enabled toggles the module as enabled or disabled, by default modules are enabled
spec.modules.logs	object	(No Description)
spec.modules.logs.enabled	boolean	Enabled toggles the module as enabled or disabled, by default modules are enabled
spec.name	string	Name of the service being deployed
spec.replicas	integer	(No Description)
spec.size	string	(No Description)
spec.tlsSecretName	string	(No Description)
spec.version	string	Version of the csv deploying
spec.zenserviceName	string	name of the zenservice cr that is used for this deployment

status

Description: (No Description)

Type: object

Table 1. Description of BaseUI/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

BundleManifest/v1alpha1

BundleManifest is the Schema for the bundlemanifests API

spec

Description: BundleManifestSpec defines the desired state of BundleManifest

Type: object

Table 1. Description of BundleManifest/v1alpha1 spec

Property	Type	Description
spec	object	BundleManifestSpec defines the desired state of BundleManifest
spec.instanced	object	GitAppSpec defines the desired state of GitApp
spec.instanced.authSecret	object	AuthSecret is the name of the secret containing credentials for Basic-Auth in the username and password keys
spec.instanced.authSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.instanced.branch	string	Branch to checkout
spec.instanced.certSecret	object	CertSecret is the name of the secret containing the ca.crt needed to access the repo if the origin uses a self-signed certificate
spec.instanced.certSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.instanced.components	array	(No Description)
spec.instanced.repo	string	Repo containing deployment files
spec.prereqs	object	GitAppSpec defines the desired state of GitApp
spec.prereqs.authSecret	object	AuthSecret is the name of the secret containing credentials for Basic-Auth in the username and password keys
spec.prereqs.authSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.prereqs.branch	string	Branch to checkout
spec.prereqs.certSecret	object	CertSecret is the name of the secret containing the ca.crt needed to access the repo if the origin uses a self-signed certificate
spec.prereqs.certSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.prereqs.components	array	(No Description)
spec.prereqs.repo	string	Repo containing deployment files

status

Description: CommonStatus that can be used with StatusUpdater

Type: object

Table 1. Description of BundleManifest/v1alpha1 status

Property	Type	Description
status	object	CommonStatus that can be used with StatusUpdater
status.conditions	array	(No Description)
status.observedGeneration	integer	(No Description)
status.phase	string	(No Description)
status.resources	object	Resources contained in status

BundleManifest/v1beta1

BundleManifest is the Schema for the bundlemanifests API

spec

Description: BundleManifestSpec defines the desired state of BundleManifest

Type: object

Table 1. Description of BundleManifest/v1beta1 spec

Property	Type	Description
spec	object	BundleManifestSpec defines the desired state of BundleManifest
spec.instanced	object	GitAppSpec defines the desired state of GitApp
spec.instanced. authSecret	object	AuthSecret is the name of the secret containing credentials for Basic-Auth in the username and password keys
spec.instanced. authSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.instanced. branch	string	Branch to checkout
spec.instanced. certSecret	object	CertSecret is the name of the secret containing the ca.crt needed to access the repo if the origin uses a self-signed certificate
spec.instanced. certSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.instanced. components	array	(No Description)
spec.instanced. repo	string	Repo containing deployment files
spec.prereqs	object	GitAppSpec defines the desired state of GitApp
spec.prereqs. authSecret	object	AuthSecret is the name of the secret containing credentials for Basic-Auth in the username and password keys

Property	Type	Description
spec.prereqs.authSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.prereqs.branch	string	Branch to checkout
spec.prereqs.certSecret	object	CertSecret is the name of the secret containing the ca.crt needed to access the repo if the origin uses a self-signed certificate
spec.prereqs.certSecret.name	string	Name of the referent. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names TODO: Add other useful fields. apiVersion, kind, uid?
spec.prereqs.components	array	(No Description)
spec.prereqs.repo	string	Repo containing deployment files

status

Description: CommonStatus that can be used with StatusUpdater

Type: object

Table 1. Description of BundleManifest/v1beta1 status

Property	Type	Description
status	object	CommonStatus that can be used with StatusUpdater
status.conditions	array	(No Description)
status.observedGeneration	integer	(No Description)
status.phase	string	(No Description)
status.resources	object	Resources contained in status

ConnectorUI/v1

For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#).

spec

Description: CommonSpecs - specs that are common to determining a deployment.

Type: object

Table 1. Description of ConnectorUI/v1 spec

Property	Type	Description
spec	object	CommonSpecs - specs that are common to determining a deployment.

Property	Type	Description
spec.configurati on	object	Additional configuration for the UI deployment
spec.configurati on.topologyMod el	string	TopologyModel - Specify the terminology to use in CP4WAIOps when defining collections of topology resource groups. Valid values are "application" or "service". If this setting is not specified, the default value will be "application".
spec.container	object	ContainerSpec - possible configurations around containers
spec.container.c ontainerPort	integer	(No Description)
spec.container.e nv	array	(No Description)
spec.container.r esources	object	ResourceRequirements describes the compute resource requirements.
spec.container.r esources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.r esources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.r esources.reques ts	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.image	object	ImageSpec - specification around how to pull an image.
spec.image.path	string	(No Description)
spec.image.pull Policy	string	PullPolicy describes a policy for if/when to pull a container image
spec.image.pull Secret	string	(No Description)
spec.license	object	License - license spec determines whether a license has been accepted
spec.license.acc ept	boolean	License acceptance switch
spec.name	string	Name of the service being deployed
spec.replicas	integer	(No Description)
spec.size	string	(No Description)
spec.tlsSecretN ame	string	(No Description)
spec.version	string	Version of the csv deploying
spec.zenservice Name	string	name of the zenservice cr that is used for this deployment

status

Description: (No Description)

Type: object

Table 1. Description of ConnectorUI/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

InsightsUI/v1

For documentation regarding install parameters, see [IBM Documentation](#). By installing this product you accept the [license terms](#).

spec

Description: (No Description)

Type: object

Table 1. Description of InsightsUI/v1 spec

Property	Type	Description
spec	object	(No Description)
spec.asm	object	Contact details for ASM
spec.asm.bindingSecret	string	Data store binding secret
spec.configuration	object	Additional configuration for the UI deployment
spec.configuration.topologyModel	string	TopologyModel - Specify the terminology to use in CP4WAI Ops when defining collections of topology resource groups. Valid values are "application" or "service". If this setting is not specified, the default value will be "application".
spec.container	object	ContainerSpec - possible configurations around containers
spec.container.containerPort	integer	(No Description)
spec.container.datarouter	object	(No Description)
spec.container.datarouter.containerPort	integer	(No Description)
spec.container.datarouter.env	array	(No Description)
spec.container.datarouter.image	object	ImageSpec - specification around how to pull an image.

Property	Type	Description
spec.container.datarouter.image.path	string	(No Description)
spec.container.datarouter.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.datarouter.image.pullSecret	string	(No Description)
spec.container.datarouter.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.datarouter.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.datarouter.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.datarouter.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.datarouter.tlsSecretName	string	(No Description)
spec.container.env	array	(No Description)
spec.container.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.uid	object	(No Description)
spec.container.uid.containerPort	integer	(No Description)

Property	Type	Description
spec.container.ubi.env	array	(No Description)
spec.container.ubi.image	object	ImageSpec - specification around how to pull an image.
spec.container.ubi.image.path	string	(No Description)
spec.container.ubi.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.ubi.image.pullSecret	string	(No Description)
spec.container.ubi.resources	object	ResourceRequirements describes the compute resource requirements.
spec.container.ubi.resources.claims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.ubi.resources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.ubi.resources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.ubi.tlsSecretName	string	(No Description)
spec.container.ubiApi	object	(No Description)
spec.container.ubiApi.containerPort	integer	(No Description)
spec.container.ubiApi.env	array	(No Description)
spec.container.ubiApi.image	object	ImageSpec - specification around how to pull an image.
spec.container.ubiApi.image.path	string	(No Description)
spec.container.ubiApi.image.pullPolicy	string	PullPolicy describes a policy for if/when to pull a container image
spec.container.ubiApi.image.pullSecret	string	(No Description)
spec.container.ubiApi.resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
spec.container.u iApi.resources.cl aims	array	Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate. This field is immutable. It can only be set for containers.
spec.container.u iApi.resources.li mits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iApi.resources.r equests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
spec.container.u iApi.tlsSecretNa me	string	(No Description)
spec.demodatas ets	array	(No Description)
spec.elasticSear ch	object	Contact details for ElasticSearch
spec.elasticSear ch.bindingSecre t	string	Data store binding secret
spec.image	object	ImageSpec - specification around how to pull an image.
spec.image.path	string	(No Description)
spec.image.pull Policy	string	PullPolicy describes a policy for if/when to pull a container image
spec.image.pull Secret	string	(No Description)
spec.kafka	object	Contact details for Kafka
spec.kafka.bindi ngSecret	string	Data store binding secret
spec.license	object	License - license spec determines whether a license has been accepted
spec.license.acc ept	boolean	License acceptance switch
spec.name	string	Name of the service being deployed
spec.postgres	object	Contact details for postgres
spec.postgres.bi ndingSecret	string	Data store binding secret
spec.replicas	integer	(No Description)
spec.size	string	(No Description)
spec.tlsSecretN ame	string	(No Description)
spec.version	string	Version of the csv deploying
spec.zenservice Name	string	name of the zenservice cr that is used for this deployment

status

Description: (No Description)

Type: object

Table 1. Description of InsightsUI/v1 status

Property	Type	Description
status	object	(No Description)
status.conditions	array	(No Description)
status.managedResources	array	(No Description)
status.phase	string	(No Description)
status.versions	object	(No Description)
status.versions.reconciled	string	(No Description)

Installation/v1alpha1

Leverage advanced, explainable AI to resolve incidents faster, gain visibility into critical workloads, and intelligently manage key application resources. [Learn More](#). By installing this product you accept the [license terms](#).

spec

Description: For detailed information about the following settings, see the IBM Documentation at https://ibm.biz/cp4aiops_470_install

Type: object

Table 1. Description of Installation/v1alpha1 spec

Property	Type	Description
spec	object	For detailed information about the following settings, see the IBM Documentation at https://ibm.biz/cp4aiops_470_install
spec.automationFoundation	object	AutomationFoundation contains parameters that can be used to customize how the IBM Automation Foundation objects are configured.
spec.automationFoundation.automationBase	string	AutomationBase is the name of the IBM Automation Foundation "AutomationBase" CR instance that the Cloud Pak should use. If a name is not specified, and no "AutomationBase" instances exist in the namespace, the Cloud Pak will attempt to provision one.
spec.automationFoundation.automationUIConfig	string	AutomationUIConfig is the name of the IBM Automation Foundation "AutomationUIConfig" CR instance that the Cloud Pak should use. If a name is not specified, and no "AutomationUIConfig" instances exist in the namespace, the Cloud Pak will attempt to provision one.
spec.automationFoundation.enabled	boolean	The Enabled flag controls whether the Cloud Pak will register with IBM Automation Foundation. The default value is true . When the flag is set to false , the Cloud Pak will not create any Automation Foundation objects, including those referenced in other parts of the AutomationFoundation spec.

Property	Type	Description
spec.enableConnectionModule	boolean	Use the Secure Tunnel to safely connect to remote data sources.
spec.groupProfiles	object	GroupProfiles contains settings for group profile features.
spec.groupProfiles.enabled	boolean	Enabled, if set to true, turns on the group profiles features.
spec.imagePullSecret	string	Select the image pull secret to use for the Cloud Pak. If no secret is selected, the global pull secret will be used.
spec.installPlanApproval	string	Approval is the user approval policy for an InstallPlan. It must be one of "Automatic" or "Manual".
spec.license	object	License acceptance is required to create resources.
spec.license.accept	boolean	I accept the product license after reading it at https://ibm.biz/cp4aiops-470-license
spec.pakModules	array	Pak Modules are modular pieces that may be enabled for deployment
spec.proxyServer	string	If the target cluster is air-gapped, specify the address and port number of an HTTP proxy that will provide access to the Slack or MS Teams service used for ChatOps, for example, proxy.company.com:8080.
spec.resourceOverrides	string	Specify the name of a configmap that finely tunes the capabilities and capacity of an installation.
spec.size	string	Select a size to determine the amount of virtual resources requested and the level of high availability support.
spec.storageClasses	string	Select a file storage class that supports ReadWriteMany access mode.
spec.storageClassLargeBlock	string	Select a storage class with a large block size (for example, 64k).
spec.topologyModel	string	Specify the terminology to use when defining collections of resource groups. You can select application or service.
spec.zones	array	Define the names of zones for a multi-zone cluster.

status

Description: InstallationStatus defines the observed state of Installation

Type: object

Table 1. Description of Installation/v1alpha1 status

Property	Type	Description
status	object	InstallationStatus defines the observed state of Installation
status.componentnamespace	string	(No Description)
status.componentstatus	object	(No Description)
status.customProfileConfigmap	string	(No Description)
status.customProfileValidationStatus	string	(No Description)
status.failureMessage	string	(No Description)

Property	Type	Description
status.failureReason	string	(No Description)
status.imagePullSecret	string	(No Description)
status.licenseacceptance	string	LicenseAcceptanceStatus describes the status of the license acceptance
status.locations	object	(No Description)
status.locations.cloudPakUiUrl	string	Launch the Cloud Pak
status.locations.csAdminHubUrl	string	Manage services and processes
status.phase	string	InstallationPhase describes the phase of the overall Installation
status.size	string	(No Description)
status.storageclass	string	(No Description)
status.storageclasslargeblock	string	(No Description)

Tunnel/v1

Tunnel is the Schema for the tunnels API

spec

Description: SecureTunnelSpec defines the desired state of Tunnel

Type: object

Table 1. Description of Tunnel/v1 spec

Property	Type	Description
spec	object	SecureTunnelSpec defines the desired state of Tunnel
spec.ApiServerURL	string	(No Description)
spec.apiReplicas	integer	(No Description)
spec.apiServerURL	string	(No Description)
spec.applicationMappingPortRange	string	(No Description)
spec.auditReceiver	object	(No Description)
spec.auditReceiver.mongodb	array	(No Description)
spec.auditReceiver.syslog	array	(No Description)
spec.auditReceivers	array	(No Description)

Property	Type	Description
spec.commonServiceNamespace	string	(No Description)
spec.controllerReplicas	integer	(No Description)
spec.defaultTunnelWorkerResources	object	(No Description)
spec.defaultTunnelWorkerResources.limits	object	(No Description)
spec.defaultTunnelWorkerResources.limits.cpu	string	(No Description)
spec.defaultTunnelWorkerResources.limits.memory	string	(No Description)
spec.defaultTunnelWorkerResources.requests	object	(No Description)
spec.defaultTunnelWorkerResources.requests.cpu	string	(No Description)
spec.defaultTunnelWorkerResources.requests.memory	string	(No Description)
spec.iamEndpoint	string	(No Description)
spec.imageDigest	string	(No Description)
spec.imageRepo	string	(No Description)
spec.imageTag	string	(No Description)
spec.ingressClass	string	(No Description)
spec.ingressDomain	string	(No Description)
spec.runAsStandAlone	boolean	(No Description)
spec.enableEgressNetworkPolicy	boolean	(No Description)
spec.enableAdminLogin	boolean	(No Description)
spec.connectionTLSCertDurationTime	string	(No Description)
spec.inventory	object	(No Description)

Property	Type	Description
spec.inventory.inventoryPort	integer	(No Description)
spec.inventory.inventoryService	string	(No Description)
spec.inventory.namespace	string	(No Description)
spec.inventory.searchApiSecrets	string	(No Description)
spec.jwtPublicURL	string	(No Description)
spec.kubernetes.Evn	boolean	(No Description)
spec.license	object	Opt-in license acceptance is required to create resources
spec.license.accept	boolean	(No Description)
spec.noManagementUI	boolean	(No Description)
spec.pullSecret	string	(No Description)
spec.runType	string	(No Description)
spec.showInConsole	boolean	(No Description)
spec.showInContainerLog	boolean	(No Description)
spec.size	string	(No Description)
spec.tunnelWorkerResources	object	////
spec.tunnelWorkerResources.limits	object	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/
spec.tunnelWorkerResources.requests	object	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. More info: https://kubernetes.io/docs/concepts/configuration/manage-compute-resources-container/
spec.uiImageDigest	string	(No Description)
spec.uiImageRepo	string	(No Description)
spec.uiImageTag	string	(No Description)
spec.uiReplicas	integer	(No Description)
spec.useTag	boolean	(No Description)
spec.version	string	(No Description)
spec.workerReplicas	integer	(No Description)

Property	Type	Description
spec.zenCoreCertificateName	string	(No Description)

status

Description: SecureTunnelStatus defines the observed state of Tunnel

Type: object

Table 1. Description of Tunnel/v1 status

Property	Type	Description
status	object	SecureTunnelStatus defines the observed state of Tunnel
status.conditions	array	(No Description)
status.versions	object	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run <code>operator-sdk generate k8s</code> to regenerate code after modifying this file Add custom validation using kubebuilder tags: https://book-v1.book.kubebuilder.io/beyond_basics/generating_crd.html
status.versions.description	string	(No Description)
status.versions.reconciled	string	(No Description)
status.versions.status	string	(No Description)
status.versions.tenantID	string	(No Description)

Infrastructure Automation custom resource definitions

Use this information to learn about the custom resource definitions (CRDs) used by Infrastructure Automation.

In the following table, CRDs that do not have a hyperlink must not be manually edited unless otherwise instructed, and are manipulated only by the operator.

Table 1. Infrastructure Automation CRDs

Kind	API version	Operator name	Description
IAConfig	v1alpha1	ibm-infrastructure-automation-operator	Build, deploy, and manage VMs and Kubernetes clusters. Enable self-service orchestration of resources across clouds. Learn More .
ManageService	v1alpha1	ibm-management-cam-install	ManageService is the Schema for the manageservices API

Kind	API version	Operator name	Description
ManageIQ	v1alpha1	ibm-management-im-install	ManageIQ is the Schema for the manageiqs API
IMInstall	v1alpha1	ibm-management-im-install	IMInstall is the Schema for the iminstalls API

IAConfig/v1alpha1

Build, deploy, and manage VMs and Kubernetes clusters. Enable self-service orchestration of resources across clouds. [Learn More](#).

spec

Description: IAConfigSpec defines the desired state of IAConfig

Type: object

Table 1. Description of IAConfig/v1alpha1 spec

Property	Type	Description
spec	object	IAConfigSpec defines the desired state of IAConfig
spec.imagePullSecret	string	Select a secret to serve as the ImagePullSecret
spec.infraAutoComposableComponents	array	Enable Infrastructure management (im) and/or Manage services (cam).
spec.license	object	License acceptance is required to create resources.
spec.license.accept	boolean	I accept the product license after reading it at https://ibm.biz/cp4aiops-470-license
spec.skipAutomationBaseStatus	boolean	This field is reserved for support person to use.
spec.skipAutomationUiConfigStatus	boolean	This field is reserved for support person to use.
spec.storageClass	string	Select a storage class that supports ReadWriteMany access mode.
spec.storageClassLargeBlock	string	Select a storage class with a large block size (for example, 64k).

Important: The following fields are deprecated and should not be used in Infrastructure Automation 4.7.0:

- skipAutomationBaseStatus
- skipAutomationUiConfigStatus

status

Description: IAConfigStatus defines the observed state of IAConfig

Type: object

Table 2. Description of IAConfig/v1alpha1 status

Property	Type	Description
status	object	IAConfigStatus defines the observed state of IAConfig
status.failureMessage	string	INSERT ADDITIONAL STATUS FIELD - define observed state of cluster Important: Run "make" to regenerate code after modifying this file
status.phase	string	InstallationPhase describes the phase of the overall Installation
status.storageclass	string	(No Description)
status.storageclasslargeblock	string	(No Description)
status.version	string	(No Description)

ManageService/v1alpha1

ManageService is the Schema for the manageservices API.

spec

Description: ManageServiceSpec defines the desired state of ManageService.

Type: object

Table 1. Description of IAManageService spec

Property	Type	Description	Default

Property	Type	Description	Default
camAPI	object	Configuration for Cloud Automation Manager API microservices	<pre>{ "camComposerApi": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camComposerUi": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camIaaS": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camMongo": { "resources": { "limits": { "cpu": "500m", "memory": "4Gi" }, "requests": { "cpu": "100m", "memory": "2Gi" } }, "camOrchestration": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camPortalUi": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camProviderTerraformApi": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "terraformSSHDir": { "terraformSSHDirSecret": "" }, "camSecret": { "secretName": "cam-api-secret" }, "camServiceLibraryUi": { "resources": { "limits": { "cpu": "300m", "memory": "512Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camServiceLibraryUiAPI": { "resources": { "limits": { "cpu": "100m", "memory": "512Mi" }, "requests": { "cpu": "50m", "memory": "256Mi" } }, "camTenantApi": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "camUiBasic": { "resources": { "limits": { "cpu": "500m", "memory": "1500Mi" }, "requests": { "cpu": "100m", "memory": "256Mi" } }, "certificate": { "certName": "cert", "replicaCount": 1 } } } } } } }</pre>
camAPI.replicaCount	int	Number of replicas for cam microservices	1
camController.replicaCount	int	Number of replicas for Controller	1

Property	Type	Description	Default
camLoggingPolicies	object	Configuration for the logging level used with Cloud Automation Manager microservices	<code>{"logLevel":"info"}</code>
camLoggingPolicies.logLevel	string	Log Level for logging	<code>"info"</code>
camMongoPV	object	MongoDB Persistent Storage	<code>{"name":"cam-mongo-pv","persistence":{"accessMode":"ReadWriteOnce","enabled":true,"existingClaimName":"","size":"15Gi","storageClassName":"","useDynamicProvisioning":false}}</code>
camMongoPV.persistence.accessMode	string	Access Mode for Storage	<code>"ReadWriteOnce"</code>
camMongoPV.persistence.enabled	bool	Enable persistence	<code>true</code>
camMongoPV.persistence.existingClaimName	string	Specify the name of the Existing Claim to be used by your application, empty string means don't use an existClaim	<code>""</code>
camMongoPV.persistence.size	string	MongoDB PVC size (in Gi)	<code>"15Gi"</code>
camMongoPV.persistence.storageClassName	string	Specify the name of the StorageClass empty string means don't use a StorageClass	<code>""</code>
camMongoPV.persistence.useDynamicProvisioning	bool	useDynamicProvisioning for persistence	<code>false</code>
camProviderTerraform	object	Configuration for Cloud Automation Manager Provider Terraform	<code>{"isolateRuntime":false,"replicaCount":1,"runtime":{"replicaCount":0},"terraformSSHDir":{"terraformSSHDirSecret":""}}</code>

Property	Type	Description	Default
camProviderTerraform.isolateRuntime	bool	Select to enable isolation mode for Provider Terraform runtime. In this mode, the runtime runs in separate pods and have read only access to the cam-terraform-pv.	false
camProviderTerraform.replicaCount	int	Number of replicas for camProviderTerraformApi	1
camProviderTerraform.runtime	object	Configuration for Cloud Automation Manager Provider Terraform runtime	{"replicaCount":0}
camProviderTerraform.runtime.replicaCount	int	Number of replicas for Provider Terraform Runtime. This value is only applicable when camProviderTerraform.isolateRuntime is enabled	0
camProviderTerraform.terraformSSHDir	object	Cloud Automation Manager Provider Terraform SSH Directory	{"terraformSSHDirSecret":""}
camProviderTerraform.terraformSSHDir.terraformSSHDirSecret	string	Secret name for Cloud Automation Manager Provider Terraform SSH Directory Contents	""

Property	Type	Description	Default
camProxy	object	CAM Proxy configuration	<code>{"customNginxSSLCertificate": {"customNginxSSLCertName": ""}, "limits": {"cpu": "500m", "memory": "1500Mi"}, "replicaCount": 1, "requests": {"cpu": "100m", "memory": "256Mi"}, "resources": null}</code>
camProxy.customNginxSSLCertificate.customNginxSSLCertName	string	Nginx SSL Custom Certificate Secret Name	<code>""</code>
camProxy.limits.cpu	string	The CPU limit	<code>"500m"</code>
camProxy.limits.memory	string	The memory limit	<code>"1500Mi"</code>
camProxy.replicaCount	int	Number of replicas for Proxy	<code>1</code>
camProxy.requests.cpu	string	The requested CPU	<code>"100m"</code>
camProxy.requests.memory	string	The requested memory	<code>"256Mi"</code>
camUI	object	UI Configuration	<code>{"camUISecret": {"secretName": "cam-ui-secret", "sessionKey": "opsConsole.sid", "sessionTTL": "43200"}, "replicaCount": 1}</code>
camUI.camUISecret	object	User Interface Secret	<code>{"secretName": "cam-ui-secret", "sessionKey": "opsConsole.sid", "sessionTTL": "43200"}</code>
camUI.camUISecret.secretName	string	User interface Secret Name	<code>"cam-ui-secret"</code>
camUI.camUISecret.sessionKey	string	Session Key	<code>"opsConsole.sid"</code>
camUI.camUISecret.sessionTTL	string	Session timeout in seconds	<code>"43200"</code>
camUI.replicaCount	int	Number of replicas for UI	<code>1</code>
database	object	Optionally use an external MongoDB	<code>{"bundled": true}</code>
database.bundled	bool	Use internal MongoDB. Set to false for external. For more information, see Using an external MongoDB in the IBM Knowledge Center	<code>true</code>
license	object	License acceptance	<code>{"accept": true}</code>

Property	Type	Description	Default
license.accept	bool	Accept the license to use the product"	true
proxy	object	Configuration for a proxy used with Cloud Automation Manager. Supply details in the secure values secret.	{"useProxy": false}
proxy.useProxy	bool	Select to use a proxy with CAM and fill out the fields in the Secure Values secret (see the Knowledge Center install instructions for more details)	false
resources	object	Amount of memory and CPU to assign to Cloud Automation Manager microservices	{"limits": {"cpu": "500m", "memory": "1500Mi"}, "requests": {"cpu": "100m", "memory": "256Mi"}}
resources.limits.cpu	string	The memory limit	"500m"
resources.limits.memory	string	The CPU limit	"1500Mi"
resources.requests.cpu	string	The requested CPU	"100m"
resources.requests.memory	string	The requested memory	"256Mi"
roks	bool	Select to indicate it is Red Hat OpenShift on IBM Cloud System	false
roksRegion	string	Red Hat OpenShift on IBM Cloud Storage Region selection	""

Property	Type	Description	Default
roksZone	string	Red Hat OpenShift on IBM Cloud Storage Zone selection	""

IMInstall/v1alpha1

IMInstall is the Schema for the iminstalls API

spec

Description: IMInstallSpec defines the desired state of IMInstall

Type: object

Table 1. Description of IMInstall/v1alpha1 spec

Property	Type	Description
spec	object	IMInstallSpec defines the desired state of IMInstall
spec.appAnnotations	object	Optional Annotations to apply to the Httpd, Kafka, Memcached, Orchestrator and PostgreSQL Pods
spec.appName	string	Application name used for deployed objects (default: iminstall)
spec.applicationDomain	string	Domain name for the external route. Used for external authentication configuration
spec.applicationImage	string	Image string used for the infrastructure management application deployment. By default it is based on the olm.relatedImage.application annotation
spec.backupLabelName	string	This label will be applied to essential resources that need to be backed up (default: manageiq.org/backup)
spec.baseWorkerImage	string	Image string used for the base worker deployments By default this is determined by the orchestrator pod
spec.databaseRegion	string	Database region number (default: 0)
spec.databaseSecret	string	Secret containing the database access information, content generated if not provided (default: postgresql-secrets)
spec.databaseVolumeCapacity	string	Database volume size (default: 50Gi)
spec.deployMessagingService	boolean	Deprecated: Flag to indicate if Kafka and Zookeeper should be deployed (default: false)
spec.enableApplicationLocalLogin	boolean	Flag to allow logging into the application without SSO (default: false)
spec.enableSSO	boolean	Flag to enable SSO in the application (default: false)
spec.enforceWorkerResourceConstraints	boolean	Flag to trigger worker resource constraint enforcement (default: true)

Property	Type	Description
spec.httpdAuthConfig	string	Secret containing the httpd configuration files Mutually exclusive with the OIDCClientSecret and OIDCProviderURL if using openid-connect
spec.httpdAuthenticationType	string	Type of httpd authentication (default: internal) Options: internal, external, active-directory, saml, openid-connect Note: external, active-directory, and saml require an httpd container with elevated privileges
spec.httpdCpuLimit	string	Httpd deployment CPU limit (default: 1000m)
spec.httpdCpuRequest	string	Httpd deployment CPU request (default: 50m)
spec.httpdImage	string	Image string used for the httpd deployment (default: os.Getenv("HTTPD_IMAGE"))
spec.httpdImageNamespace	string	Image namespace used for the httpd deployment (default: docker-namespace-public.artifactory.swg-devops.com/hyc-cp4mcm-team-docker-local/infra-mgmt) Note: the exact image will be determined by the authentication method selected
spec.httpdImageTag	string	Image tag used for the httpd deployment (default: 2-el9)
spec.httpdMemoryLimit	string	Httpd deployment memory limit (default: 200Mi)
spec.httpdMemoryRequest	string	Httpd deployment memory request (default: 100Mi)
spec.imagePullSecret	string	Secret containing the image registry authentication information needed for the manageiq images
spec.initialAdminGroupName	string	Group name to create with the super admin role. This can be used to seed a group when using external authentication
spec.internalCertificatesSecret	string	Secret containing all of the necessary certificates to secure communication between pods (default: internal-certificates-secret)
spec.kafkaCpuRequest	string	Kafka deployment CPU request (default: 250m)
spec.kafkaCpuLimit	string	Kafka deployment CPU limit (default: 250m)
spec.kafkaImage	string	Image string used for the kafka deployment (default: os.Getenv("KAFKA_IMAGE"))
spec.kafkaImageName	string	Image used for the kafka deployment (default: docker.io/bitnami/kafka)
spec.kafkaImageTag	string	Image tag used for the kafka deployment (default: latest)
spec.kafkaMemoryLimit	string	Kafka deployment memory limit (default: 1024Mi)
spec.kafkaMemoryRequest	string	Kafka deployment memory request (default: 256Mi)
spec.kafkaSecret	string	Secret containing the kafka access information, content generated if not provided (default: kafka-secrets)
spec.kafkaVolumeCapacity	string	Kafka volume size (default: 1Gi)
spec.license	object	Opt-in license acceptance is required to create resources
spec.license.accept	boolean	Accept is an opt-in license acceptance required to deploy resources

Property	Type	Description
spec.memcachedCpuLimit	string	Memcached deployment CPU limit (default: 1000m)
spec.memcachedCpuRequest	string	Memcached deployment CPU request (default: 50m)
spec.memcachedImage	string	Image string used for the memcached deployment (default: :)
spec.memcachedImageName	string	Image used for the memcached deployment (default: manageiq/memcached)
spec.memcachedImageTag	string	Image tag used for the memcached deployment (default: 1-el9)
spec.memcachedMaxConnections	string	Memcached max simultaneous connections (default: 1024)
spec.memcachedMaxMemory	string	Memcached item memory in megabytes (default: 64)
spec.memcachedMemoryLimit	string	Memcached deployment memory limit (default: 200Mi)
spec.memcachedMemoryRequest	string	Memcached deployment memory request (default: 100Mi)
spec.memcachedSlabPageSize	string	Memcached max item size (default: 1m, min: 1k, max: 1024m)
spec.migrationsRan	array	A list of CR data migrations that have been run
spec.oidcAuthIntrospectionURL	string	URL for OIDC authentication introspection Only used with the openid-connect authentication type. If not specified, the operator will attempt to fetch its value from the "introspection_endpoint" field in the Provider metadata at the OIDCProviderURL provided.
spec.oidcCaCertSecret	string	Secret containing the trusted CA certificate file(s) for the OIDC server. Only used with the openid-connect authentication type
spec.oidcClientSecret	string	Secret name containing the OIDC client id and secret Only used with the openid-connect authentication type
spec.oidcProviderURL	string	URL for the OIDC provider Only used with the openid-connect authentication type
spec.orchestratorCpuLimit	string	Orchestrator deployment CPU limit (default: 1000m)
spec.orchestratorCpuRequest	string	Orchestrator deployment CPU request (default: 200m)
spec.orchestratorImage	string	Image string used for the orchestrator deployment (default: / :)
spec.orchestratorImageName	string	Image name used for the orchestrator deployment (default: infra-mgmt-orchestrator)
spec.orchestratorImageNamespace	string	Image namespace used for the orchestrator and worker deployments (default: manageiq)
spec.orchestratorImageTag	string	Image tag used for the orchestrator and worker deployments (default: latest-18.y)

Property	Type	Description
spec.orchestratorInitialDelay	string	Number of seconds to wait before starting the orchestrator liveness check (default: 2400)
spec.orchestratorMemoryLimit	string	Orchestrator deployment memory limit (default: 2048Mi)
spec.orchestratorMemoryRequest	string	Orchestrator deployment memory request (default: 1024Mi)
spec.postgresqlCpuLimit	string	PostgreSQL deployment CPU limit (default: 10000m)
spec.postgresqlCpuRequest	string	PostgreSQL deployment CPU request (default: 500m)
spec.postgresqlImage	string	Image string used for the postgresql deployment (default: :)
spec.postgresqlImageName	string	Image used for the postgresql deployment (Default: docker.io/manageiq/postgresql)
spec.postgresqlImageTag	string	Image tag used for the postgresql deployment (Default: 13-el9)
spec.postgresqlMaxConnections	string	PostgreSQL maximum connection setting (default: 1000)
spec.postgresqlMemoryLimit	string	PostgreSQL deployment memory limit (default: 8192Mi)
spec.postgresqlMemoryRequest	string	PostgreSQL deployment memory request (default: 2048Mi)
spec.postgresqlSharedBuffers	string	PostgreSQL shared buffers setting (default: 1GB)
spec.serverGuid	string	Server GUID (default: auto-generated)
spec.storageClassName	string	StorageClass name that will be used by manageiq data stores
spec.tlsSecret	string	Secret containing the tls cert and key for the ingress, content generated if not provided (default: tls-secret)
spec.uiWorkerImage	string	Image string used for the UI worker deployments By default this is determined by the orchestrator pod
spec.opentofuRunnerImage	string	Image string used for the Opentofu runner deployments By default this is determined by the orchestrator pod
spec.webserverWorkerImage	string	Image string used for the webserver worker deployments By default this is determined by the orchestrator pod
spec.zookeeperCpuRequest	string	Zookeeper deployment CPU request (default: 250m)
spec.zookeeperCpuLimit	string	Zookeeper deployment CPU limit (default: 250m)
spec.zookeeperImage	string	Image string used for the zookeeper deployment (default: :)
spec.zookeeperImageName	string	Image used for the zookeeper deployment (default: docker.io/bitnami/zookeeper)
spec.zookeeperImageTag	string	Image tag used for the zookeeper deployment (default: latest)
spec.zookeeperMemoryLimit	string	Zookeeper deployment memory limit (default: 256Mi)

Property	Type	Description
spec.zookeeperMemoryRequest	string	Zookeeper deployment memory request (default: 256Mi)
spec.zookeeperVolumeCapacity	string	Zookeeper volume size (default: 1Gi)

status

Description: IMInstallStatus defines the observed state of IMInstall

Type: object

Table 1. Description of IMInstall/v1alpha1 status

Property	Type	Description
status	object	IMInstallStatus defines the observed state of IMInstall

Accessing APIs

You can access and use IBM Cloud Pak® for AIOps APIs after you obtain an access token for authentication.

About this task

To use any IBM Cloud Pak for AIOps API, you must first authenticate your credentials by including an access token in your API calls. This token is required because a password-based authentication approach is not supported for API calls with IBM Cloud Pak for AIOps, for instance, you cannot call an LDAP server to directly authenticate passwords.

The token that is generated is scoped to your user profile and the permissions that you are assigned when the access token is generated. With your access token, you can use API to access everything that you would typically be able to access when you log in to the IBM Cloud Pak for AIOps console.

To get a Platform UI access token, first you must have either an Identity and Access Management (IAM) access token or an API key that is authenticated by your user credentials. Then, you can use the IAM access token or API key to obtain your Platform UI access token. This Platform UI access token must be included in the authorization header of your calls to IBM Cloud Pak for AIOps API:

To create your token, complete either of the following procedures based on your preferred method:

- [Get access token by using your API key](#)
- [Get access token by using your username and password](#)
- [Get authentication by using Platform UI API Key](#)

Important: You can directly access the Swagger UI for only the Topology API and Metric API.

- For more information about accessing Metric service Swagger UI, see [Accessing Metric service Swagger UI](#).
- For more information about accessing topology service Swagger UI, see [Accessing Topology service Swagger UI](#).

Get access token by using your API key

1. Generate your platform API key:

1. Log in to the IBM Cloud Pak for AIOps console to authenticate your user credentials.

For more information, such as for how to obtain the URL route for accessing the Cloud Pak for AIOps console, see [Accessing the Cloud Pak for AIOps console](#).

2. From the toolbar, click your avatar.
3. Click **Profile and settings**.
4. Click **API key > Generate new key**.
5. Click **Generate**.
6. Click **Copy** and save your key somewhere safe. You cannot recover this key if you lose it.

If you lose your API key, repeat the preceding steps to generate a new API key. Your old API key becomes invalid, and any applications or scripts cannot authenticate to the platform until you provide your new API key. For more information, see [Generating API keys for authentication](#).

2. Get your Platform UI access token by running the following cURL command:

```
curl -k -X POST \
  https://<hostname>/icp4d-api/v1/authorize \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "<username>",
    "api_key": "<API_key>"
  }'
```

- **<hostname>** - The IBM Cloud Pak platform (**cpd**) route hostname.
- **<username>** - Your username.
- **<API_key>** - The API key that you generated in the previous step.

The API response can resemble the following sample:

```
{
  "_messageCode_": "200",
  "message": "Success",
  "token": "<encoded-token-string>"
}
```

- **<encoded-token-string>** - Your Platform UI access token.

Get access token by using your username and password

1. Get your Identity and Access Management (IAM) access token with the following cURL command:

```
curl -k -X POST -H "Content-Type: application/x-www-form-
urlencoded; charset=UTF-8" \
-d "grant_type=password&username=<username>&password=<password>&scope=openid" \
https://<cluster_address>/idprovider/v1/auth/identitytoken
```

Where:

- **<username>** - Your username.

- **<password>** - Your password.
- **<cluster_address>** - The IBM Cloud Pak for AIOps console route. For more information about how to get this Cloud Pak for AIOps console route, see [Accessing the Cloud Pak for AIOps console](#).

The command returns an **access_token**, **refresh_token**, and **id_token**, as shown in the following example:

```
{ "access_token":
"eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f0005d21
7e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a0
9e380feefba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0
c4af818b7f7d6c32e21465909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb797999
5d7044c7b9cb05945c91b63a938703641e0fded339fb4c22e2383743a94a30c41892804193744
e0c0f020909f9579555bf691b240fafb558f76877fe0cb88ecbb3266fedbc7c541129270f6778
4d11ed658998b536841e0fdcf50a9ad056d2cabf717cb13326e4f620a6ce172d8da4701b820c5
ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3328351e3c
38a8b199040067ca5837169ce62a88282d1c8551d762fbdbff77727cb51dd62213cef58dfb88e3
04abbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c9e0e56b104e12bf1
ea3764faeac258fe1c3e896da412607a71ad8b4224efcfa0eafbc15f5e7af5b8baa41163c2204
19c7249e9652ca7b5692b42cbe4c7d88c18d77440ec350582f51880e7354eed76ebd8ce760b27
a6ca5808c6fc51ef843ee5d98e5bbafaeb4096301853fa5fdc876275defd3ecabd0eb656c7cd2
441e523e0c5468a1f261fb44",
"token_type": "Bearer", "expires_in": 43199, "scope": "openid"
"refresh_token": "6q4griAg9yCiGINQvF0Dp7N9hqXhcXZrAsqWWYgl6XQ80Uexsq",
"id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjoiYWRRmZDc4MmEw
OTclZTNmMzc2ZTkxZTI3YjJkNTYxZmQ0OTNiNTQzMISInJlYWxtTmFtZSI6ImN1c3RvbVJlYWxtI
iwidW5pcXVlU2VjdXJpdHl0YW1lIjoieYWRtaW4iLCJpc3MiOiJodHRwczovL215Y2xlc3Rlci5pY3
A6OTQ0My9vaWRjL2VuZHBvaW50L09QIiwiYXVkJjoieMGQzYzA3MTc5OTYxYmEzMWEyODY5NDU0NDQ
wM2E0NDYiLCJleHAiOiJlNTQ5MTQ2NTIsImhhdCI6MTU1NDg4NTglMiwic3ViIjoieYWRtaW4iLCJ0
ZW50IjoiZm9udC51bGhjbmdzIjpbXX0.CnT0qWECpJR9R16W-
IOqrXjSJR8DelRsDUXcX6hy_I0DPQ7hU55Bhcq6UChEg3qiWWRBkWrFIxikXPjEjw2B9oziEd8U8A
EO-4LEaXOp5Lk1shvyxBQFDDgyUwgYGb-
erRbO_Sl1K4xotutLg4nhoydwTXs71Zn7GC4UW8j1qkh1bFe5iLgKidCZsjyPo-
2GNYEQn0ufHH3KCR4DkHi6GX2RUxisNecwDzN19P5JSyjlS-
r5QUZJ0b0DytKuY5HxpswpIFa09U8JlYAFoOZ18eO_CzERHRQ_IilePmagGAK-
eLJjmcNqY1zynfpEUuKlWUR5rVGHGzSbGA8J4CLvg" }
```

From this example, the following content is the **access token** that is needed for obtaining a platform UI access token:

```
"access_token":
"eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f0005d21
7e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a0
9e380feefba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0
c4af818b7f7d6c32e21465909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb797999
5d7044c7b9cb05945c91b63a938703641e0fded339fb4c22e2383743a94a30c41892804193744
e0c0f020909f9579555bf691b240fafb558f76877fe0cb88ecbb3266fedbc7c541129270f6778
4d11ed658998b536841e0fdcf50a9ad056d2cabf717cb13326e4f620a6ce172d8da4701b820c5
ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3328351e3c
38a8b199040067ca5837169ce62a88282d1c8551d762fbdbff77727cb51dd62213cef58dfb88e3
04abbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c9e0e56b104e12bf1
ea3764faeac258fe1c3e896da412607a71ad8b4224efcfa0eafbc15f5e7af5b8baa41163c2204
19c7249e9652ca7b5692b42cbe4c7d88c18d77440ec350582f51880e7354eed76ebd8ce760b27
a6ca5808c6fc51ef843ee5d98e5bbafaeb4096301853fa5fdc876275defd3ecabd0eb656c7cd2
441e523e0c5468a1f261fb44"
```

2. Get your Platform UI access token by running the following cURL command:

```
curl -k X GET 'https://<hostname>/v1/preauth/validateAuth' \
-H 'username: admin' \
-H 'iam-token: <iam-token>'
```

- **<hostname>** - The IBM Cloud Pak platform (**cpd**) route hostname.
- **<username>** - Your username.
- **<iam-token>** - The IAM access token that you obtained from the response in the previous step.

The API response can resemble the following sample:

```
{
  "_messageCode_": "200",
  "message": "Success",
  "token": "<encoded-token-string>"
}
```

- **<encoded-token-string>** - Your Platform UI access token.

Get authentication by using Platform UI API Key

1. Generate your platform API (Zen) key:

1. Log in to the IBM Cloud Pak for AIOps console to authenticate your user credentials.

For more information, such as for how to obtain the URL route for accessing the Cloud Pak for AIOps console, see [Accessing the Cloud Pak for AIOps console](#).

2. From the toolbar, click your avatar.

3. Click **Profile and settings**.

4. Click **API key > Generate new key**.

5. Click **Generate**.

6. Click **Copy** and save your key somewhere safe. You cannot recover this key if you lose it.

If you lose your API key, repeat the preceding steps to generate a new API key. Your old API key becomes invalid, and any applications or scripts cannot authenticate to the platform until you provide your new API key. For more information, see [Generating API keys for authentication](#).

2. Generate a base64 format of your Platform UI API Key:

```
ZENAPIKEY=$(echo "<username>:<API_key>" | base64 -w 0)
```

- **<username>** - Your username.
- **<API_key>** - The API key that you generated in the previous step.

3. The output is an encoded value. See the following example.

```
YWRtaW46MUFGKblRIYngxNFJUVnBoR0FTdW0wcDJEOFl6UXFyV1lmU2tmeW5qYgo=
```

This encoded (base64 format) Platform UI API Key is used as the value for **<ZenApiKey>**.

4. Construct your Authorization header with the Platform UI API Key.

```
--header "Authorization: ZenApiKey <ZenApiKey>"
```

- **<ZenApiKey>** - The Platform UI API key that you get in the previous step.

5. Run the following cURL command:

```
curl "https://<Endpoint URL>" --header 'Content-Type: application/json' --
header "Authorization: ZenApiKey <ZenApiKey>" --header 'X-TenantID: cfd95b7e-
3bc7-4006-a4a8-a73a79c71255' --insecure
```

- **<Endpoint URL>** - The Endpoint URL as specified on the API page of the service that you want to use, for example, you can find the Endpoint URL for metrics [here](#).

Next steps

Access IBM Cloud Pak for AIOps API with your Platform UI authentication token or API keys. Include the Platform UI access token in the authorization header of your API calls.

For more information on the available API and related Swagger documentation, see [APIs](#).

Generating API keys for authentication

With an API key, you can automatically authenticate to the platform or to a specific instance of a service from a script or application. Your API keys are associated with your credentials and are specific to you. With API keys, you can authenticate without entering your username and password.

You can generate:

- [A platform API key](#)

Your platform API key enables scripts and applications to access everything that you would typically be able to access when you log in to the console.

Before you begin

Before you generate an API key, consider the following factors:

What level of access is required?

As a security best practice, it is recommended that you give the least amount of access necessary to the script or application. For example, if a script needs to interact only with a specific service instance, generate an API key for that service instance.

Important: Be sure to store your API keys somewhere safe. If you lose an API key, you'll need to create a new key and update any scripts or applications where the key is used.

Platform API key

With a platform API key, you can access everything that you would typically be able to access when you log in to the console.

To generate a platform API key:

1. Log in to the console.
2. From the toolbar, click your avatar.
3. Click **Profile and settings**.
4. Click **API key** > **Generate new key**.
5. Click **Generate**.
6. Click **Copy** and save your key somewhere safe. You cannot recover this key if you lose it.

If you lose your API key, repeat the preceding steps to generate a new API key. Your old API key becomes invalid, and any applications or scripts cannot authenticate to the platform until you provide your new API key.

If you believe that your API key is compromised:

1. Log in to the console.
2. From the toolbar, click your avatar.
3. Click **Profile and settings**.
4. Click **API key > Revoke current key**.
5. Click **Revoke**.

Any applications or scripts that use the key won't be able to authenticate to the platform.

Platform UI APIs

Learn more about the Platform UI APIs that are available for you to use to manage users and the personalization of the console.

[User management APIs](#)

Create and manage users and access to the console with the user management API.

[Custom cards API](#)

Create custom cards to display information on the console home page with the Custom cards REST API.

Creating custom home page cards with API

Create custom cards to display information on the console home page with the Custom cards REST API.

For example, if you want to show disk usage as a percentage of available disk space, you can add a custom card to the home page that displays the data in a donut or bar chart.

You can use the Custom cards REST API to complete the following tasks:

- [Generate an authorization token](#)
- [Get a list of custom cards](#)
- [Create or replace a custom card](#)
- [Edit a custom card](#)
- [Delete a custom card](#)

You cannot use the Custom cards API to replace, edit, or delete existing cards that are added by default.

Sample custom cards

You can start with a template to create each of the custom cards that are shown in the following image. Review the [template types](#) for a description of the templates that are used.

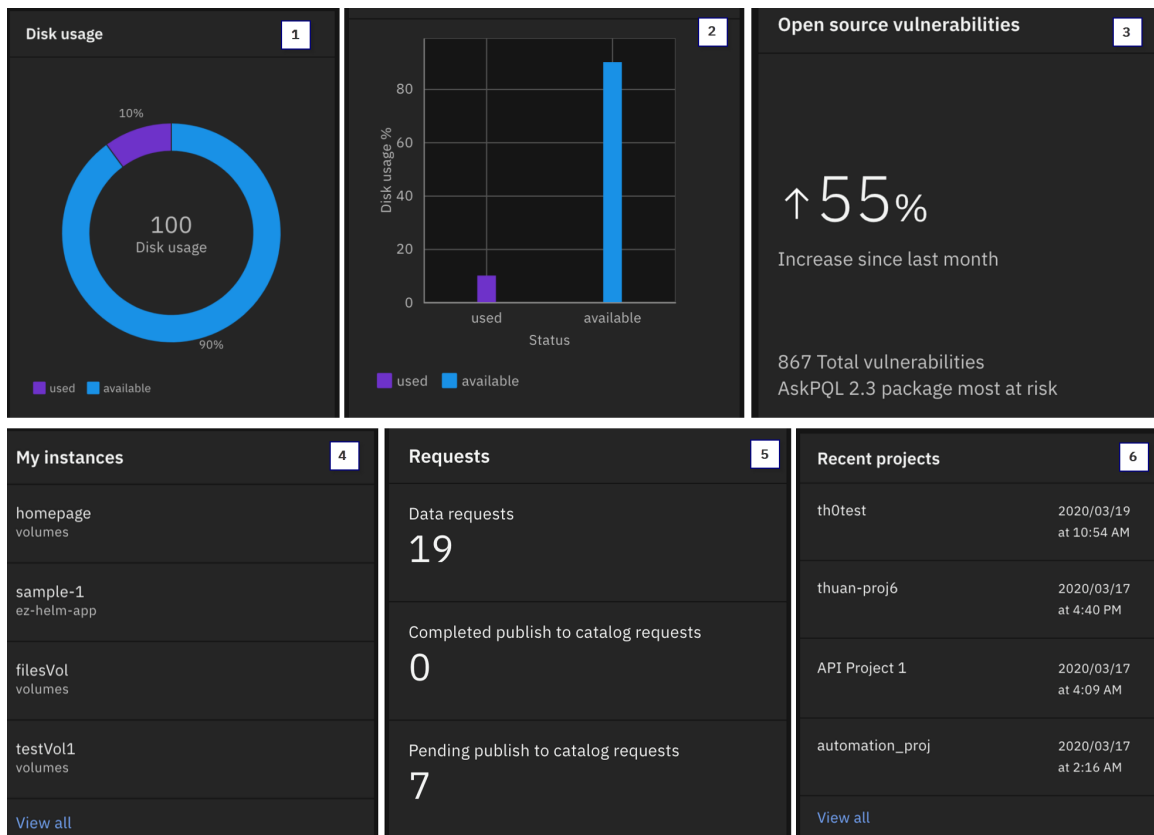


Figure 1. Custom cards types

Template types:

1. The **donut** template illustrates an array of values as percentages in a donut chart.
2. The **bar** template creates a two-dimensional bar graph.
3. The **big_number** template highlights significant data.
4. The **text_list** displays a list of text strings.
5. The **number_list** displays a list of numeric values and associated text.
6. The **list** displays a list of navigation links. The list can contain headers and associated clickable links.

For the JSON payload that describes these custom cards, see the sample command in [Create or replace a custom card](#).

Generate an authorization token

To use the Custom cards REST API, you must first authenticate. Use the token that you get in the response as a token in subsequent API calls.

1. Get an Identity Management (IM) access token:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" \
-d "grant_type=password&username=<username>&password=<password>&scope=openid" \
https://<cluster_address>/idprovider/v1/auth/identitytoken
```

- **<username>** is your username.
- **<password>** is your password.
- **<cluster_address>** is the IBM Cloud Pak console route. You can get the IBM Cloud Pak console route by running the following command:

```
oc get route -n <your-foundational-services-namespace> cp-console -o
jsonpath='{.spec.host}'
```

Following is a sample output:

```
cp-console.apps.mycluster.mydomain.com
```

Based on the example output, your console URL would be `https://cp-console.apps.mycluster.mydomain.com`.

The username and password are defined in the `config.yaml` file.

2. Use the IM access token to call the Platform UI token API:

```
curl -k X GET 'https://<master_node_hostname>/v1/preauth/validateAuth' \
-H 'username: admin' \
-H 'iam-token: <iam-token>'
```

- `<master_node_hostname>` the hostname for the master node.
- `<iam-token>` is the IM token that you obtained from the response in the previous step.

Get a list of custom cards

Retrieves the properties for the custom cards that you can update.

Curl

```
GET /zen-data/v1/custom_cards
```

Sample command

```
curl -kiv -X GET \
  https://my-deployment-url/zen-data/v1/custom_cards \
  -H 'Authorization: Bearer authorization-token' \
  -H 'Content-Type: application/json'
```

Sample response

In the following code block, the body of the response is replaced with . . .

```
{
  "_messageCode_": "success",
  "message": "Successfully retrieved custom cards",
  "requestObj": {
    ...
  }
}
```

Create or replace a custom card

Creates a custom card or replaces an existing custom card.

You must include the card definition as a JSON string in the HTTP request body. The definition specifies the structure and the content of the card.

Table 1. Card parameters

Parameter	Type	Description

Parameter	Type	Description
key	String	The name of the card that you want to create or replace. The string must be unique. If a custom card with the specified key does not exist, the custom card is created, otherwise the existing custom card is replaced.
data	object	Provides data to the card. You must provide either a data_url object or a data object. The data object is an object that populates data in the card. The properties of the object must match the type of template defined. The expected structure for each template_type parameter is in following examples.
data_url	endpoint	Provides data to the card. You must provide either a data_url object or a data object. The data_url object is an endpoint that is called to populate data in the card. The response that is sent by the endpoint matches the type of template defined. The expected response structure for each template_type parameter is shown in the following examples.
description	String	Helps users understand the purpose of the card. The string must be 200 characters or less.
drilldown_url (Optional)	String	Defines the URL that the user navigates to when they click View details . If this parameter is not defined, the card does not display a View details link.
order	integer	Defines the position of the card. The value must be in the range 1 - 50.
permissions	array	Specifies the permissions that a user needs to see the card. The array must have at least one permission. If the array has no permissions parameter, the card is shown to everyone.
refresh_rate (Optional)	Integer	The interval in seconds at which the content is refreshed by the source.
template_type	enumerated_type	Defines the display type of the card. The template_type parameter must be one of the following values: - donut - bar - big_number - text_list - number_list - list
title	String	The display name for the card. The string must be 40 characters or less.
window_open_target (Optional)	string	Specifies the target attribute or the name of the window. The following values are supported: - Empty or not provided - the URL gets loaded on the current page by default. - _blank - URL is loaded into a new window or tab every time. - name - The name of the window where the URL is loaded. The name does not specify the title of the new window.

Use the following examples to understand the **data** object.

```
**bar\_data**
```

```
"data": {
  "bar_data": {
    x_axis_label: 'Packages',
    y_axis_label: 'Projects',
    "data_array": [
      {
```

```

        "group": "Qty",
        "value": 65000
    },
    {
        "group": "More",
        "value": 29123
    },
    {
        "group": "Sold",
        "value": 35213
    },
    {
        "group": "Restocking",
        "value": 51213
    },
    {
        "group": "Misc",
        "value": 16932
    }
]
}
}

```

****big_number_data****

```

"data": {
    "big_number_data": {
        prefix: 'ArrowUp32',
        metric: '1.6',
        suffix: '%',
        sub_text: 'Since last month',
        footer_1: '867 Total vulnerabilities',
        footer_2: 'AskPQL 2.3 package most at risk'
    }
}

```

****donut_data****

```

"data": {
    "donut_data": {
        "center_label": "Disk usage",
        "data_array": [
            {
                "group": "2V2N 9KYPM version 1",
                "value": 20000
            },
            {
                "group": "L22I P66EP L22I P66EP L22I P66EP",
                "value": 65000
            },
            {
                "group": "JQAI 2M4L1",
                "value": 75000
            },
            {
                "group": "J9DZ F37AP",
                "value": 1200
            },
            {
                "group": "YEL48 Q6XK YEL48",
                "value": 10000
            },
        ]
    }
}

```

```

        {
            "group": "Misc",
            "value": 25000
        }
    ]
}

**list\_data**

"data": {
    list_data: {
        headers: [
            'project_name',
            'last_updated_time'
        ],
        rows: [
            {
                'project_name': 'Improving customer online registration',
                'last_updated_time': '2 min ago',
                'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-10acf78a4e71'
            },
            {
                'project_name': '2020 Q1_Increasing Search performance
and..',
                'last_updated_time': '30 min ago',
                'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-abc'
            },
            {
                'project_name': '2019 Customer shoes sales analysis',
                'last_updated_time': '1 hour ago',
                'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-def'
            },
            {
                'project_name': 'Project X Final version',
                'last_updated_time': 'Today 11:04 PM',
                'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-grs'
            },
            {
                'project_name': 'Project X Final version 2',
                'last_updated_time': 'Yesterday 12:45 AM',
                'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-tuv'
            }
        ]
    }
}

**number\_list\_data**

"data": {
    "number_list_data": {
        rows: [
            {
                drilldown_url: '/zen/#/openSource/packages',
                label: 'Total approved packages',
                value: '3'
            },
            {
                label: 'Total licenses',
                value: '2345'
            },
            {
                label: 'Total user interactions',
                value: '25K'
            }
        ]
    }
}

```

```

    }
  ]
}

**text\_list\_data**

"data": {
  "text_list_data": {
    rows: [
      {
        drilldown_url: '/zen/#/openSource/packages',
        label: 'Data virtualization',
        sub_text: 'dv'
      },
      {
        label: 'Watson assistant',
        sub_text: 'assistant'
      },
      {
        label: 'Watson openscale',
        sub_text: 'aios'
      }
    ]
  }
}

```

Use the following examples to understand the `data\_url` object. The format of the expected response that is provided in the `data\_url` from the endpoint for each card type is shown in the following examples.

****donut****

```

{
  "center_label": "Disk usage",
  "data_array": [
    {
      "group": "2V2N 9KYPM version 1",
      "value": 20000
    },
    {
      "group": "L22I P66EP L22I P66EP L22I P66EP",
      "value": 65000
    },
    {
      "group": "JQAI 2M4L1",
      "value": 75000
    },
    {
      "group": "J9DZ F37AP",
      "value": 1200
    },
    {
      "group": "YEL48 Q6XK YEL48",
      "value": 10000
    },
    {
      "group": "Misc",
      "value": 25000
    }
  ]
}

```

****bar****

```

{
  "x_axis_label": "Packages",
  "y_axis_label": "Projects",
  "data_array": [
    {
      "group": "Australia",
      "value": 250
    },
    {
      "group": "Brazil",
      "value": 120
    },
    {
      "group": "Canada",
      "value": 190
    },
    {
      "group": "Germany",
      "value": 170
    }
  ]
}

**big\_number**

{
  prefix: 'ArrowUp32',
  metric: '1.6',
  suffix: '%',
  sub_text: 'Since last month',
  footer_1: '867 Total vulnerabilities',
  footer_2: 'AskPQL 2.3 package most at risk'
}

**text\_list**

{
  rows: [
    {
      label: 'Data virtualization',
      sub_text: 'dv'
    },
    {
      label: 'Watson assistant',
      sub_text: 'assistant'
    },
    {
      label: 'Watson openscale',
      sub_text: 'aios'
    }
  ]
}

**number\_list**

{
  rows: [
    {
      label: 'Total approved packages',
      value: '3'
    },
    {
      label: 'Total licenses',
      value: '2345'
    }
  ]
}

```

```

    },
    {
      label: 'Total user interactions',
      value: '25K'
    }
  ]
}

**list**

{
  headers: [
    'project_name',
    'last_updated_time'
  ],
  rows: [
    {
      'project_name': 'Improving customer online registration',
      'last_updated_time': '2 min ago',
      'nav_url': '/projects/49753ad5-dcf5-4f94-913f-10acf78a4e71'
    },
    {
      'project_name': '2020 Q1 Increasing Search performance',
      'last_updated_time': '30 min ago',
      'nav_url': '/zen/#/mydata/datasets'
    },
    {
      'project_name': '2019 Customer shoes sales analysis',
      'last_updated_time': '1 hour ago',
      'nav_url': '/zen/#/mydata/datasets'
    },
    {
      'project_name': 'Project X Final version',
      'last_updated_time': 'Today 11:04 PM',
      'nav_url': '/zen/#/mydata/datasets'
    },
    {
      'project_name': 'Project X Final version 2',
      'last_updated_time': 'Yesterday 12:45 AM',
      'nav_url': '/zen/#/mydata/datasets'
    }
  ]
}

```

Curl

```
PUT /zen-data/v1/custom_cards/{key}
```

Sample command

```

curl -k -X PUT 'https://my-deployment-url/zen-data/v1/custom_cards/unique-card-key' \
--header 'Authorization: Bearer authorization-token' \
--header 'Content-Type: application/json' \
--data-raw '{
  "permissions": ["manage_catalog"],
  "order": 1,
  "title": "Disk usage",
  "template_type": "big_number",
  "data": {
    "big_number_data": {
      "metric": "55",
      "sub_text": "Increase since last month",
      "prefix": "ArrowUp32",

```



```

        "suffix": "%",
        "footer_1": "867 Total vulnerabilities",
        "footer_2": "AskPQL 2.3 package most at risk"
    }
}

```

Sample response

```

{
  "messageCode": "success",
  "data": [
    {
      "id": "537163840404062200",
      "name": "os_vulnerabilities",
      "title": "Open source vulnerabilities",
      "description": "",
      "drilldown_url": "",
      "order": 12,
      "template_type": "big_number",
      "data_url": "",
      "empty_state": {},
      "source_url": "",
      "refresh_rate": 0,
      "data": {
        "big_number_data": {
          "metric": "55",
          "sub_text": "Increase since last month",
          "prefix": "ArrowUp32",
          "suffix": "%",
          "footer_1": "867 Total vulnerabilities",
          "footer_2": "AskPQL 2.3 package most at risk"
        }
      }
    }
  ],
  "message": "PUT custom card succeeded"
}

```

Edit a custom card

Edit the information that is displayed on an existing card.

You can edit the following content:

- data
- data_url
- description
- drilldown_url
- order
- permissions
- template_type
- title

To edit a custom card, use the rules that are specified in [Create or replace a custom card](#).

Curl

```
PATCH /zen-data/v1/custom_cards/{key}
```

Path parameters

Table 2. Path parameters - patch

Parameter	Type	Description
key	String	The name of the card that you want to update.

Sample command

This command updates the title of a custom card called `disk_usage_donut`.

```
curl -kiv -X PATCH \
  https://my-deployment-url/zen-data/v1/custom_cards/disk_usage_donut \
  -H 'Authorization: Bearer authorization-token' \
  -d '{
    "title": "Disk usage"
  }'
```

Sample response

```
{
  "_messageCode_": "success",
  "data": [
    {
      "id": "1",
      "name": "homepage_card_disk_usage",
      "title": "Disk usage",
      "description": "",
      "drilldown_url": "/zen/#/v1/diskUsage",
      "order": 2,
      "template_type": "donut",
      "data_url": "/zen-data/v1/homepage/diskUsage",
      "empty_state": {},
      "source_url": "",
      "refresh_rate": 0,
      "data": null
    }
  ],
  "message": "Successfully edited custom card"
}
```

Delete a custom card

Deletes an existing card with the specified key.

Curl

```
DELETE /zen-data/v1/custom_cards/{key}
```

Path parameters

Table 3. Path parameters - delete

Parameter	Type	Description
key	String	The name of the custom card that you want to delete.

Sample command

This command deletes the `os_vulnerabilities` card.

```
curl -kiv -X DELETE \
  https://my-deployment-url/zen-data/v1/custom_cards/os_vulnerabilities \
  -H 'Authorization: Bearer authorization-token'
```

Sample response

```
{
  "_messageCode_": "success",
  "message": "Successfully deleted custom card"
}
```

IM APIs

Identity management (IM) APIs.

- [OIDC Registration APIs](#)
- [Identity Provider V3 APIs](#)

Preparing to run API commands

Before you run API commands, retrieve the authentication token and download the CA certificate for your cluster.

Procedure

1. Set the project to the namespace where you deployed foundational services.

```
oc project <your-foundational-services-namespace>
```

2. Retrieve the authentication tokens. You can run curl commands.

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" -d "grant_type=password&username=`oc get secret platform-auth-idp-credentials -o json | jq -r .data.admin_username | base64 -d`&password=`oc get secret platform-auth-idp-credentials -o json | jq -r .data.admin_password | base64 -d`&scope=openid" https://`oc get routes cp-console -o jsonpath='{.spec.host}'`/idprovider/v1/auth/identitytoken | jq
```

The command returns an `access_token`, `refresh_token`, and `id_token`, as shown in the following example:

```
{
  "access_token": "eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f0005d217e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a09e380feefba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0c4af818b7f7d6c32e21465909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb7979995d7044c7b9cb05945c91b63a938703641e0fded339fb4c22e2383743a94a30c41892804193744e0c0f020909f9579555bf691b240fafb558f76877fe0cb88ecbb3266fedbc7c541129270f67784d11ed658998b536841e0fdcf50a9ad056d2cabf717cb13326e4f620a6ce172d8da4701b820c5ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3328351e3c38a8b199040067ca5837169ce62a88282d1c8551d762fbdf77727cb51dd62213cef58dfb88e304abbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c",
  "refresh_token": "eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f0005d217e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a09e380feefba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0c4af818b7f7d6c32e21465909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb7979995d7044c7b9cb05945c91b63a938703641e0fded339fb4c22e2383743a94a30c41892804193744e0c0f020909f9579555bf691b240fafb558f76877fe0cb88ecbb3266fedbc7c541129270f67784d11ed658998b536841e0fdcf50a9ad056d2cabf717cb13326e4f620a6ce172d8da4701b820c5ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3328351e3c38a8b199040067ca5837169ce62a88282d1c8551d762fbdf77727cb51dd62213cef58dfb88e304abbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c",
  "id_token": "eb837eaf32459b711945a9d2259880119056e805ff0d2f36421cc171f94e58fef349f0005d217e36889b62271d9f00fc7cb5b1fe5a86546d9dee8e22bca39b8f90d6cb61dae7fc383447823a09e380feefba5bea0c994408470a49db0df32ddc2b0cca9381519e60a63daae9b87ebfe9400b0c4af818b7f7d6c32e21465909efc8aa02804808f23ff96ac342b3b1c35230ac8858dbcb7979995d7044c7b9cb05945c91b63a938703641e0fded339fb4c22e2383743a94a30c41892804193744e0c0f020909f9579555bf691b240fafb558f76877fe0cb88ecbb3266fedbc7c541129270f67784d11ed658998b536841e0fdcf50a9ad056d2cabf717cb13326e4f620a6ce172d8da4701b820c5ffe23223e7fb5725b244a1dd45538a0c7ca09a643759aaa2d8585a28689cae968ab3328351e3c38a8b199040067ca5837169ce62a88282d1c8551d762fbdf77727cb51dd62213cef58dfb88e304abbc48063246b7e9f39650a0ac86f6c72973b702b79faf34b68afb9412c"
}
```


token value from "eb837e to 1fb44". You can access IBM Cloud Pak for AIOps user management APIs by specifying the access token in the request header. Run the following command, where **<Access token>** is the following displayed access token:

```
export ACCESS_TOKEN=<Access token>
```

5. Obtain a copy of the CA certificate for your cluster. If you can access the boot node, the CA certificate file is `/<installation_directory>/cluster/cfc-certs/root-ca/ca.crt`.

OIDC Registration APIs

APIs to manage authentication.

Base path: `https://<cluster_address>/idprovider/v1/auth/`

Note: The `/idauth` route is removed and not supported for client registration.

Client Registration API

Following is the curl command to register the API:

```
curl -i -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" \
--header "Content-Type: application/json" \
--data "@platform-oidc-registration.json" \
https://<cluster_address>/idprovider/v1/auth/registration
```

You can also use the access token in the body:

```
curl -i -k -X POST -d "access_token=$ACCESS_TOKEN" \
https://<cluster_address>/idprovider/v1/auth/registration
```

The contents of the platform-oidc-regisration.json file are in the following example:

```
{
  "token_endpoint_auth_method": "client_secret_basic",
  "client_id": <WLP_CLIENT_ID>,
  "client_secret": <WLP_CLIENT_SECRET>,
  "scope": "openid profile email",
  "grant_types": ["authorization_code", "client_credentials", "implicit",
    "refresh_token", "urn:ietf:params:oauth:grant-type:jwt-bearer"],
  "response_types": ["code", "token", "id_token token"],
  "application_type": "web",
  "subject_type": "public",
  "post_logout_redirect_uris": ["https://<ICP_PROXY_IP>:
<PORT_WHERE_SERVICE_RUNS>"],
  "preauthorized_scope": "openid profile email general",
  "introspect_tokens": true,
  "trusted_uri_prefixes": ["https://<ICP_ENDPOINT>:8443",
"https://<ICP_PROXY_IP>"],
  "redirect_uris": ["https://<ICP_PROXY_IP>:
<PORT_WHERE_SERVICE_RUNS>/auth/liberty/callback"],
}
```

The `WLP_CLIENT_ID` and `WLP_CLIENT_SECRET` must be generated by the user who is trying to register the client and the values must be unique.

Note: Calling `/userinfo` does not work for the tokens with `grant_type` as `client_credentials`. Use `/instrospect` endpoints instead.

Delete the client ID

1. Export these variables:

```
export CLIENT_ID=<client_id>
```

2. Run the curl command to delete the ID.

```
curl -i -k -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" \
https://icp-cluster-ip:8443/idprovider/v1/auth/registration/$CLIENT_ID
```

Call the authorization endpoint to display the login page

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/authorize

Command

GET

Command output format

application/json

The sample command resembles the following code:

```
GET https://<cluster_address>/idprovider/v1/auth/authorize?
client_id=$oauth_client_id&redirect_uri=https://$http_host/auth/liberty/callback&r
esponse_type=code&scope=openid+email+profile&state=$request_uri;
```

Get access token by using username and password

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP


```
YFcBplFGwp9FDZb27A"
}
```

Note: All the idprovider APIs are rate limited. If rate limit exceeds, it throws **429 error**. Standard rate limits: 100 requests per 10 seconds and burst of 20.

Get access token by using client_credentials

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/token

Command

POST

Command output format

application/json

Following is the curl command:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" \
-d "grant_type=client_credentials&client_id=<oidc_client_ID>&client_secret=
<oidc_client_secret>&scope=openid" \
https://<cluster_address>/idprovider/v1/auth/token --insecure
```

Note: You can also use `https://<cluster_address>/idprovider/v1/auth/identitytoken`

The sample response resembles the following code:

```
{
  "access_token":
"38400d87f39a7c328a4605265eb601bebd9426e2ef6f1b51a449da6a9cb08e03543857ac4ffbd7d2c
259867c89324563c5a89f026683aca9262858ae7ffb1e635242eabab3d579793e8f9da09070708dccf
2a8d660f3be06550f02af681d2fa64562fb9dc3df1b19839a5d3933311f89348634fa6908fa7d2d505
84ffd36f9dc298a3411d3f5abad5c7f45283428ecf0de249eac5534136c31317493f85363126bfe9a6
f582403c34a3dab96e3e7bba83c263f1a4ff8d8609fca888852e097e3bc382b822576a53e55e6753c5
7f79d5703cf6b6bed4b015702ce3ce1636fd834944231fa77eb90079bca398be511f22fd58792a3766
a100af10f274e6b9d75a2be2fe6ab18a3ce2ed0c8da7542e0b79f08e32a9ddced6a389572e6247230e
1b62adf5fb0ee6549c06f99b85afc7cccd7a51012dea5df40fc27a934be37e9465ddb46a4f43ec542f
aebc4e6dd062189392b802b8a0ad8c38a00a14f7b9625bcdab251b87cd478c0e5d3c79f8887797da76
7f209f5fb2b3d44c8b051f49c2ed680a14cd15388b545ca573540184bb27be28378dbe0ecbe2d71c0a
c3d7365fce5f1948ead1576f444f70c87d7ba89352b0d2b795a11ccc5ad06441c4143a3e78f80316c7
```



```
2110ba7062159f249719c664818befd6514b1526498729fe624852128495a5fa9c57ba8c9386a1040e
0bb8013e93a751722de6e85966994cefce4c43066",
  "token_type": "Bearer",
  "expires_in": 43199,
  "scope": "openid",
}
```

Get access token by using cpclient_credentials

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/token

Command

POST

Command output format

application/json

Following is the curl command:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
-d "grant_type=cpclient_credentials&client_id=<client_ID>&client_secret=
<client_secret>&scope=openid" \
https://<cluster_address>/idprovider/v1/auth/token --insecure
```

The sample response resembles the following code:

```
{
  "access_token":
"3b17b831c5f21e46c64718d26fe415ce2fc2f5482121ab8f213a28db8f3838c6a6cda5e3a9a8d38be
359cb3c8351655e7fac2af40a805fb42e701d0ad2a28ddf92421a282165d3d1ba7b2dc56c163e907c1
ba9662b2b5ea30eb29bf28b6d999bb73deff665c12320ee22aacbedd13ba86d53467887cfee9d7962f
6968906d2ab8f7b716a521cfbdc9b2820c424c1609f933a5240b0c637c6e4ef76007abbe6dd6e0f41e
5d4f2867ca9afe6efb3160d9573b4d18ef02eb044c86e3faf0f9095a7989e1bb756f2e862781e8600b
50854f648a3cde34c9359a26596c42096fc2d30d6742885d8c5d5eb4ac9a81ff130f255d304e4b6f63
041983ac5897a2941dcb9e69d1537c176c94854b70e761bfbefcf77aba5fff3f397fcddce3f317e49f
148740d659d807b51f52757c5c5a0fd23fcf8bbd397fefe6c63e850c641eb458791194c2c8eb0652ba
373bab41b95651778f8185924d7c2ec9c9584e8cbbac256884feb4028a92d5e1bb569775b5b5362f8a
5a58d154d98d5158fda494e067e96d91aeea3e88ea6768e7a9aa1da273bc9282c168e57c826ffcf8b3
01967260eadab33d98c656fd756fcc206f6e5b659f1d6f844cb52c4008d2aa5a94ea86c2448fdb2947
c7d2d507690e5ab7d08fb463c2471b52fbff5149bc21f1169ba0e52daa125cb859c9f574297814ccd3
a8eef8b5c92f5f890a3831ef3c42bfa0ae98d457d",
}
```

```
{
    "token_type": "Bearer",
    "expires_in": 43199,
    "scope": "openid",
    "id_token":
        "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6ImNPeXZidk5sQXhKUWk2dnZ5OU5jRGJHUkh3RW5VvNJRbWphUVlCWjRYOTQifQ.eyJ0eYXNoIjoieyZllzM5ODY2YmNmM2VlNTgwOGQ2Zjc1ODdkZWZiZTgwYzEyZGUyZCIsInJlYWxtImFtZSI6ImNlc3RvbVJlYWxtIiwiaWidw5pcXVlU2VjdXJpdHl0YW11IjoieY3A0YXV0b19jbGlbnQilJCjpc3MiOiJodHRwciovLzE5Ny4wLWljAuMT0NDMvaWRhdXRoL29pZGMvZW5kcG9pbmQvTlAiLCJhdWQiOiJjcDRhdXRvX2NsaWVudCIsImV4cCI6MTYyMDIwMjM4NywiawWF0IjoxNjIwMTU5MTg3LClJzdWIiOiJjcDRhdXRvX2NsaWVudCIsInByZWZlcmlJlZF9lc2VybmFtZSI6ImNwNGFlDG9fY2xpZW50IiwibmFtZSI6ImNwNGFlDG9fY2xpZW50IiwiaWZGlzcGxheV9uYW11IjoieY3A0YXV0b190ZXN0ZXIifQ.ImJFI9KM-zt3yJziSolpzenagWGzh9Rdlta64KwOCeeszNLFNKWezAV_5_ShP40pkMBCza7mwCLVFCpAbFrlrIWANIG44zbBndgqYnBVmqDDIYrJuk-yyLBOSiJOCP02wn7Ltjtvt70XP_sbcP9uxsu2V7TwyITSxSpucyd58A376jfG1EJTKmsEzlC4Efde3qYy-7Dn8UMF-6QrvHUXEnSDEbxex0OJFz4NLnkJ1NspWbO5k5k4BlyPXqOAavynQVsmQ_095oFff2owFQDxQyZt7SBvg4mpe_6n2eJr21tj2Uyabhz-N7ZwNUSYxegBnO0WO9xKJDeP8tjpy0dJ5obUHCQ"}
}
```

Get information about a user

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

```
/idprovider/v1/auth/userInfo
```

Command

POST

Command output format

```
application/json
```

For information about **\$ACCESS_TOKEN**, see [Preparing to run API commands](#).

The sample curl command resembles the following code:

```
curl -k -X POST --header "Authorization: Bearer $ACCESS_TOKEN" \
https://<cluster address>/idprovider/v1/auth/userInfo
```

You can also use the access token in the body:

```
curl -k -X POST -d "access_token=$ACCESS_TOKEN" \
https://<cluster address>/idprovider/v1/auth/userInfo
```

The response resembles the following code:

```
{
  "sub": "admin",
  "iss": "https://mycluster.icp:9443/idprovider/v1/auth"
}
```

Call introspect endpoint

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/introspect

Command

GET

Command output format

application/json

1. Export these variables. To get the values, see [Client Registration API](#).

```
export TOKEN=<your access token here>
export CLIENT_ID=<client_id here>
export CLIENT_SECRET=<client_secret here>
```

2. Get the Basic authorization header by using the following command:

```
BASIC_AUTH_HEADER=`echo -n "$CLIENT_ID:$CLIENT_SECRET" | base64 -w 0`
```

3. Run the curl command to call the endpoint.

```
curl -H "Authorization: Basic $BASIC_AUTH_HEADER" -d "token=$TOKEN"
https://<cluster_address>/idprovider/v1/auth/introspect
```

The response resembles the following code:

```
{
  "sub": "admin",
  "grant_type": "resource_owner",
  "realmName": "customRealm",
  "scope": "openid",
  "uniqueSecurityName": "admin",
  "iss": "https://127.0.0.1:9443/idprovider/v1/auth",
  "active": true,
}
```

```
"exp": 1529358338,  
"token_type": "Bearer",  
"iat": 1529315138,  
"
```

Revoke access token or refresh token

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/revoke

Command

POST

Command output format

application/json

1. Export these variables. To get the values, see [Client Registration API](#).

```
export TOKEN=<your access token here>  
export CLIENT_ID=<client_id here>  
export CLIENT_SECRET=<client_secret here>
```

2. Get the Basic authorization header by using the following command:

```
BASIC_AUTH_HEADER=`echo -n "$CLIENT_ID:$CLIENT_SECRET" | base64 -w 0`
```

3. Run the curl command to call the endpoint.

```
curl -k -X POST -H "Authorization: Basic $BASIC_AUTH_HEADER" -d  
"token_type_hint=access_token&token=$TOKEN" \  
https://<cluster_address>/idprovider/v1/auth/revoke
```

The response resembles the following code:

```
{}
```

Get a new access token by using the refresh token

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/token

Command

POST

Command output format

application/json

To get the refresh token, see [Preparing to run API commands](#). To get the client ID and secret, see [Client Registration API](#).

The sample curl command resembles the following code:

```
curl -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
-d "grant_type=refresh_token&client_id=<client_ID>&client_secret=
<client_secret>&scope=openid&refresh_token=<refresh_token>" \
https://<cluster_address>/idprovider/v1/auth/token --insecure
```

The response resembles the following code:

```
{
  "access_token":
"77f3ea9695e50d147a3081990c331f8ce9baa0b6d02ac4e970c886eabccd7aa7e7f12e1897ceacbfd
6bdaf0881ed5a725f214209eb20b9415c2fcf4ad1afb90412a247aeab6ab0e026e08013b8f2b773b5b
db2d8d3c1247e9e7ebeaa8c9c9c66c1e85caf78105e35e934a28f21619bef2ff17cebe75792da86b4a
65c19973713559569e92ae6aa86ddb8ee48991c6ced9caf41ae6c3b88f67fcaacf8c2c6af82018b5f5
5a4e35c1b9026438b690a606de0314bdced35eab21642b4b6c33c5241db457f2564840b9d32c255d0b
fa9e4fda176416f7481c205ee98912790a11134597ce7245264669568fd69153a8e2f240df9edb4df3
b219e213c3cfb0366713802a9a525fe85c9ec2a8c54ba61b5d845054ff23eb466c990c15dcb025ef32
0f36bb21ec0d0a412bcdcafb57da6b239891e22c139a7d4057f84fd741215ed5567c3f4b824d9bbf
e92d56b77fe1712d35cea60e12f5207b727e3cc658db1b8b5002780049a5faefd8ccc2ccee9100472d
fff58978ee3e7303547dc4ea03025275e58ec4e3da8e6ae91939bfb092f1ce78fe2d91124c179f55bd
a4027957093090c4f47037771e9cacf227867063c909e9aee3bf87140426052821116c6484037822a4
1f05a0fa565276b5ff1a8a654d3d5d119f6a665469a7591e4ec197d6a90bd586b8b95e227b9869b865
4c23c10f78fc6a3fcbbe6d543638f379736193643",
  "token_type": "Bearer",
  "expires_in": 43199,
  "scope": "openid",
  "refresh_token": "5QM3H8fmGjxhPRyYlQ77s4Z5APOHVk5276ItT8q41e2xKNMxF6"
}
```

Get the OIDC configuration from the well-known configuration endpoint

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/.well-known/openid-configuration

Command

GET

Command output format

application/json

The sample curl command resembles the following code:

```
curl -k https://<cluster_address>/idprovider/v1/auth/.well-known/openid-configuration --insecure
```

The response resembles the following code:

```
{
  "introspection_endpoint": "https://<cluster_address>/idprovider/v1/auth/introspect",
  "coverage_map_endpoint": "https://<cluster_address>/idprovider/v1/auth/coverage_map",
  "issuer": "https://<cluster_address>/idprovider/v1/auth",
  "authorization_endpoint": "https://<cluster_address>/idprovider/v1/auth/authorize",
  "token_endpoint": "https://<cluster_address>/idprovider/v1/auth/token",
  "jwks_uri": "https://<cluster_address>/idprovider/v1/auth/jwk",
  "response_types_supported": ["token", "id_token", "id_token token"],
  "subject_types_supported": ["public"],
  "id_token_signing_alg_values_supported": ["RS256"],
  "userinfo_endpoint": "https://<cluster_address>/idprovider/v1/auth/userInfo",
  "registration_endpoint": "https://<cluster_address>/idprovider/v1/auth/registration",
  "scopes_supported": ["openid", "email", "profile"],
  "claims_supported": ["sub"],
  "response_modes_supported": ["query"],
  "grant_types_supported": ["client_credentials", "password", "refresh_token", "authorization_code"],
  "token_endpoint_auth_methods_supported": ["client_secret_post"],
  "display_values_supported": ["page"],
  "claim_types_supported": ["distributed"],
  "claims_parameter_supported": true,
  "request_parameter_supported": true,
  "request_uri_parameter_supported": true,
  "require_request_uri_registration": true,
  "check_session_iframe": "https://<cluster_address>/idprovider/v1/auth/check_session_iframe",
  "end_session_endpoint": "https://<cluster_address>/idprovider/v1/auth/end_session",
  "revocation_endpoint": "https://<cluster_address>/idprovider/v1/auth/revocation",
  "app_passwords_endpoint": "https://<cluster_address>/idprovider/v1/auth/app_passwords",
  "app_tokens_endpoint": "https://<cluster_address>/idprovider/v1/auth/app_tokens",
  "personal_token_mgmt_endpoint": "https://<cluster_address>/idprovider/v1/auth/personalTokenManagement",
  "users_token_mgmt_endpoint": "https://<cluster_address>/idprovider/v1/auth/usersTokenManagement",
  "client_mgmt_endpoint": "https://<cluster_address>/idprovider/v1/auth/clientManagement",
  "code_challenge_methods_supported": ["plain", "S256"]}

```

Get the Liberty and iam-token keys from the JSON Web Token (JWK) endpoint

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

/idprovider/v1/auth/jwk

Command

GET

Command output format

application/json

The sample curl command resembles the following code:

```
curl -k https://<cluster_address>/idprovider/v1/auth/jwk --insecure
```

The response resembles the following code:

```
{ "keys":
[ { "kty": "RSA", "e": "AQAB", "use": "sig", "kid": "N23lhv0Waa2mXKHDv1F37tVByN8PofD1PPouOB
X-Bq8", "alg": "RS256", "n": "o1TdyuOlHNzVNGN8S0aBWiSd7E-89k741fQ-iJd4nD3ZF1z4TbmM-
lGhr2zsbV91M_IeoJz1JZcUKevl2us2JLKOQ8bG6T58qsioxwt3AL7KJ_ap3d3sUqNQk6zWzf08_BUhizs
lmIjsJkhH9G1_Mj5vUaU8mL4k6P6SsSMklwifHrCkt0N2fPM6SvBRoRTW0En14IczEyPXLrbPpy92YLMab
PEdP0Mmv4iK6_m1uXES3HD9cpfm7LuYgqZ2Ws3NrvfM4a9FqR8OriR6tD5t4hbtT5S3UGq-
eE6vTqyJtix0uPctMRSGAx9xSWi6B-bXKRnuBATNez-FdvtMcCf_oQ" },
{ "kty": "RSA", "n": "AIdJNOnqGoCpfZcg1-
AM0OnVaQCfcZkAweku7D5uM6CVuXsdsxip_liHpTs7A01e8BM3qCxH_YbtTqbLqxR2TKmLSzGMG3QnMZzm
OunBuR_w2KuBQyz7IBDImaQlCDuEEv05wnQiryFj5B_wK6dHIRdbrlOFTP2ebjEf8gkwjxdyl32vJ-
Pqy0FksAfxHFTaccSuOrVycFLtx_MyzyexP_N76du_n6GyjkqzeUbLDdJHET4Vfdp6R4O5Cdz9zMQI4sy
7r07rFLLJMrP9rcuRZQWVKZjm4X6Cw3ptnOwVlsvEesD4W0mBHZoqjZ3Dz5ET5IGHGPuV8p39M6rDON2A
s", "e": "AQAB", "alg": "RS256", "kid": "20170721-00:00:00", "use": "sig" },
{ "kty": "RSA", "n": "AIAaIOCBGGJGXsKmW7JGLXqe7nHSWcnCMQ51gDvNhL5oAV0QxWRjpW0GDScpoA_K
CBn5zEkTqSDrxgmhWka3-Uv_D6xaeWZWRWh7Wnp3AIbc-LuL-aiKuwwWqKclmoopovT_F5-
IEfPZHN87COPEOyQ_MY-jgSuG5UBHABhMP8dEb7C35IIFqbahHKFBPgZiTwu970kAZYZJmzFw-
1PbAfSR3EHkxcduTeSWlROat1gr7CmJhNBYKUK4xpb_vt4iDWvkbikbWR_x3nsWnzntq5TjGYjejyS7rYN
FR2W-gmkxGXbMiAzq2UkbFJwTnxEp7rMuKC4CiybcWyNO-
VmDKVU", "e": "AQAB", "alg": "RS256", "kid": "20170515-00:00:00", "use": "sig" },
{ "kty": "RSA", "n": "AIkfUhuKp6NMes2A4423vhLCXNbbqshDNOa5y-GHUOxn-
rSsXJY9yBu9ykMwhcZTsWuCvtlFyo2IwQC9aBya1MJ6elebw1fYVbwG3w55ZevL19LJCzSI8vtC8yitv6x
KF1dHRWf4VHq35PCdY8gm-uOm-7OUzm7qB1NOIb8c-nhfK-
PK0cfkAVJKdpEu8ALLwG6pBBk7ZAoe3PLpABYlmy40iKEy1D0jqWpb2mamKZdLuOZ2QbhUgE5aI1eGtXYS
eIbFuFTTlhJUEKjA-iBD2mE7CFKhfkjQFvjy_jDMg-X8y9sR8jrX0sWp2Z3mYkCmc-
```

```
XgN1I_Ws5sIyTxoyOJU", "e": "AQAB", "alg": "RS256", "kid": "20170401-00:00:00", "use": "sig"}, {"kty": "RSA", "n": "AIdJNOnqGoCpfZcg1-AMOnVaQCfcZkAweku7D5uM6CVuXsdsxip_liHpTs7A01e8BM3qCxH_YbtTqbLqxR2TKmLSzGMG3QnMZzmOunBuR_w2KuBQyz7IBDImaQlCDuEEv05wnQiryFj5B_wK6dHIRdbrlOFTP2ebjEf8gkwjxdyl32vJ-Pqy0FksAfxHFTaccSuOrVycFLtx_MyzyexP_N76du_n6GyjkqzeUbLDdJHET4Vfdp6R4O5Cdz9zMQI4sy7r07rFLLJMrP9rcuRZQWVKZjM4X6Cw3ptnOwVlsvEesD4W0mBHZoqjZ3Dz5ET5IGHGWpuV8p39M6rDON2As", "e": "AQAB", "alg": "RS256", "kid": "20170301-00:00:00", "use": "sig"}, {"e": "AQAB", "kty": "RSA", "n": "o1TdyuOlHNzVNGN8S0aBWiSd7E-89k741fQ-iJd4nD3ZF1z4TbmM-lGhr2zsbV91M_IeoJzlJZcUKevl2us2JLKOQ8bG6T58qsioxwt3AL7KJ_ap3d3sUqNQk6zWzf08_BUhizslmIjsJkhH9G1_Mj5vUaU8mL4k6P6SsSMklwifHrCkt0N2fPM6SvBRoRTW0En14IczEypXLRbPpy92YLMabPEdP0Mmv4iK6_m1uXES3HD9cpfm7LuYgqZ2Ws3NrvfM4a9FqR8OriR6tD5t4hbtT5S3UGq-eeE6vTqyJtix0uPcTMRSGAx9xSWi6B-bXKRnuBATNez-FdvTMcCf_oQ", "use": "sig", "kid": "ICk0vQmsxQXvN87q-C8-2s91ts6xiifg15T0iv1KJpo"}]}}
```

Identity Provider APIs

Use Identity Provider (IdP) V3 APIs to register IdP connection and configuration.

If you are configuring SSO (single sign-on), you must use the IdP V3 APIs.

See the following notes:

- You can get the hostname from the route available in your foundational services namespace. To get the host details, see [Getting the host details](#).
- The **Administrator** IM role is allowed to access the IdP V3 APIs.

IdP V3 APIs

You can register the following providers and connections by using the IdP V3 APIs:

- OIDC provider For more information, see [Registering OIDC clients by using IdP V3](#).
- SAML provider For more information, see [Registering SAML clients by using IdP V3](#). To configure SAML, use IBM Cloud Pak for AIOps UI. For more information, see [Configuring single sign-on using IBM Cloud Pak for AIOps UI](#).
- LDAP For more information, see [Configuring LDAP connection by using IdP V3](#).

OIDC clients

Currently, only the following platforms are verified to register OIDC by using IdP V3 APIs.

Note: Service providers might change the links that are listed here. If any links is broken, check the service provider website for the new location.

- [IBM Security Verify](#)
 - **Note:** In the IBM Security Verify application, whenever you map attributes, enable the **Send all known user attributes in the ID token** option. And, do not use spaces in IBM Security Verify groups.
- [Google Cloud Platform](#)
- [Microsoft](#)
- [Okta](#)

Before you begin, register yourself in your service provider application. During registration, use the following URLs:

- For application URL, use the cp-console URL.
- For redirect URL, use `https://<cp-console-url>/ibm/api/social-login/redirect/<name of the oidc>`.

After you register, you get a unique client ID, client secret, and `discovery_url` endpoint.

Note: The curl commands that are used in the following sections contain different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

- [Registering the OIDC clients](#)
- [Deleting the registration of the client](#)
- [Getting the list of registered OIDC clients](#)
- [Updating the OIDC clients](#)

Registering OIDC clients

API details - register OIDC clients - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

idprovider/v3/auth/idsource

Command

POST

Command output format

application/json

Note: By using the POST call, you can get the unique identifier (UID) as a successful response. The UID is required for the further operations or tasks. You can also get the list of UIDs of registered IdPs by using the [Getting the list of registered OIDC clients by query](#) API.

Sample curl command to register a client

```
curl --insecure POST 'https://cp-console.apps.vij-soc.cp.fyre.ibm.com/idprovider/v3/auth/idsource/'
```

```
--header "Authorization: Bearer $ACCESS_TOKEN"
--header 'Content-Type: application/json'
--data-raw
'{
  "name": "oidc_isv",
  "description": "This is a oidc config for isv",
  "protocol": "oidc",
  "type": "IBMVerify",
  "idp_config": {
    "discovery_url": "https://bedrock-
iam.verify.ibm.com/oidc/endpoint/default/.well-known/openid-configuration",
    "client_id": "4ad41ddc-5f10-4d84-a24f-fb776607895e",
    "client_secret": "rqJMyEgee",
    "token_attribute_mappings": {
      "groups": "groupIds",
      "given_name": "given_name",
      "family_name": "family_name",
      "sub": "uid",
      "email": "email"
    }
  }
}'
```

Sample response - register a client

```
{
  "status": "success",
  "message": "Identity provider {oidc_isv} is successfully registered with
unique identifier NpiV6qBhsSDEjKRvJDQL1"
}
```

Sample error message - register a client

```
400 "error": "E11000 duplicate key error collection: platform-
db.cloudpak_ibmid_v3 index: name_1 dup key: { : <oidc name>}"
500 "error": "Other errors"
401 "error": "Insufficient user permission for role :Viewer"
202 OK Response
```

Deleting the registration of a client

API details - delete registration - DELETE

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

idprovider/v3/auth/idsource/:UID

Command

DELETE

Command output format

application/json

Sample curl command to delete the client registration

```
curl --insecure DELETE 'https://cp-console.apps.vij-soc.cp.fyre.ibm.com/idprovider/v3/auth/idsource/NpiV6qBhsSDEjKRvJDQL1' --header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - delete client registration

```
{
  "status": "success",
  "message": "{NpiV6qBhsSDEjKRvJDQL1} is deleted"
}
```

Sample error message - delete client registration

```
400 "error": "Cannot find {oidc uid}"
401 "error": "Insufficient user permission for role :Viewer"
200: Ok Response
```

Getting the list of registered OIDC clients

You can get the list of registered OIDC clients by using either of the following methods:

- [Unique identifier \(UID\)](#)
- [Query](#)

Getting the list of registered OIDC clients by UID

API details - get OIDC clients by UID - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

GET

Command output format

application/json

Sample curl command to get the list of registered clients by UID

```
curl --insecure --request GET 'https://cp-console.cp.fyre.ibm.com/idprovider/v3/auth/idsource/yOLi5oHo3t-es4aJFLkC8' --header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - list registered clients by UID

```
{
  "name": "oidc_1",
  "description": "This is a oidc config",
  "protocol": "oidc",
  "type": "IBMVerify",
  "idp_config": {
    "discovery_url": "https://bedrock-iam.verify.ibm.com/oidc/endpoint/default/.well-known/openid-configuration",
    "client_id": "My_Client_ID",
    "client_secret": "My_Client_Secret",
    "token_attribute_mappings": {
      "groups": "groupIds"
    }
  },
  "uid": "NpiV6qBhsSDEjKRvJDQL1"
}
```

Note: For security reasons, the values of the `client_secret` is not displayed in the output when you retrieve the list of registered clients with UID.

Sample error message - list registered clients by UID

```
400 "error": "Cannot find {oidc uid}"
401 "error": "Insufficient user permission for role :Viewer"
200: Ok Response
```

Getting the list of registered OIDC clients by query

API details - get OIDC clients by query - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

GET

Command output format

application/json

Sample curl command to get the list of registered clients by query

```
curl --insecure --request GET 'https://cp-console.cp.fyre.ibm.com/idprovider/v3/auth/idsource?protocol=oidc' --header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - list registered clients by query

```
{
  "idp": [
    {
      "name": "oidc_google",
      "description": "This is a oidc config for google",
      "protocol": "oidc",
      "type": "google",
      "idp_config": {
        "discovery_url": "https://accounts.google.com/.well-known/openid-configuration",
        "client_id": "My_Client_ID",
        "client_secret": "My_Client_Secret",
        "token_attribute_mappings": {
          "groups": "groupIds",
          "given_name": "firstName",
          "family_name": "lastName",
          "sub": "uid",
          "email": "email",
          "uniqueSecurityName": "uniqueSecurityName"
        }
      },
      "uid": "Uo4Y31mV7unaj5NtyfpIY"
    },
    {
      "name": "oidc_isv",
      "description": "This is a oidc config",

```

```

        "protocol": "oidc",
        "type": "IBMVerify",
        "idp_config": {
            "discovery_url": "https://bedrock-
iam.verify.ibm.com/oidc/endpoint/default/.well-known/openid-configuration",
            "client_id": "My_Client_ID",
            "client_secret": "My_Client_Secret",
            "token_attribute_mappings": {
                "groups": "groupIds",
                "given_name": "given_name",
                "family_name": "family_name",
                "sub": "uid",
                "email": "email",
                "uniqueSecurityName": "uniqueSecurityName",
                "preferred_username": "preferred_username",
                "displayName": "displayName"
            }
        },
        "uid": "NpiV6qBhsSDEjKRvJDQL1"
    }
]
}

```

Note: For security reasons, the values of the `client_secret` is not displayed in the output when you retrieve the list of registered clients with query.

Sample error message - list registered clients by query

```

400 "error": "Cannot find {oidc uid}"
401 "error": "Insufficient user permission for role :Viewer"
200: Ok Response

```

Updating the OIDC clients

API details - update OIDC clients - PUT

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster Port

Path

idprovider/v3/auth/idsource/:UID

Command

PUT

Command output format

application/json

Sample curl command to update the client

```
curl --insecure PUT 'https://cp-console.cp.fyre.ibm.com/idprovider/v3/auth/idsource/NpiV6qBhsSDEjKRvJDQL1'
--header "Authorization: Bearer $ACCESS_TOKEN"
--header 'Content-Type: application/json'
--data-raw
'{
  "name": "oidc_isv",
  "description": "This is a oidc config for isv",
  "protocol": "oidc",
  "type": "IBMVerify",
  "idp_config": {
    "discovery_url": "https://bedrock-iam.verify.ibm.com/oidc/endpoint/default/.well-known/openid-configuration",
    "client_id": "4ad41ddc-5f10-4d84-a24f-fb776607895e",
    "client_secret": "rqJMyEgee",
    "token_attribute_mappings": {
      "groups": "groupIds",
      "given_name": "given_name",
      "family_name": "family_name",
      "sub": "uid",
      "email": "email"
    }
  }
}'
```

Sample response - update client

```
{
  "status": "success",
  "message": "{NpiV6qBhsSDEjKRvJDQL1} is Updated."
}
```

Sample error message - update client

```
400 "error": "Document not found"
401 "error": "Insufficient user permission for role :Viewer"
200: Ok Response
```

SAML clients

You can register the SAML IdP by the following methods:

- [SAML with SCIM dependency registration](#)
- [SAML registration without any dependency](#)
- [SAML with LDAP dependency registration](#)

Note: You can configure only one SAML registration at an instance. The UID generated on the POST request is `defaultSP`. For `idp_metadata`, you need to provide the base64-encoded value of the generated IdP metadata `xml` file.

Get the base64-encoded value by using the following command:

```
cat idp-metadata-xml | base64
```

Note: The maximum accepted size of the metadata file is **2 MB**. If the size of the metadata file is more than 2 MB, an error message is displayed during the SAML registration.

SAML with SCIM dependency registration

You can register IdP V3 with the following platforms:

- [IdP V3 registration with IBM Security Verify](#)
- [IdP V3 registration with Okta](#)

Note: The curl commands that are used in the following sections contain different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Note: By using the POST call, you can get the unique identifier (UID) as a successful response. The UID is required for the further operations or tasks. You can also get the list of UIDs of registered IdPs by using the [Getting the list of registered OIDC clients by query](#) API.

IdP V3 registration with IBM Security Verify

The following example shows IdP V3 registration with IBM Security Verify that supports SCIM-enabled IdP with SAML. Hence, the `token_attribute_mappings` and `scim_attribute_mappings` can be different for the respective IdPs.

API details - registration with IBM Security Verify - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource

Command

POST

Command output format

application/json

Sample curl command - registration with IBM Security Verify


```

curl --insecure POST 'https://cp-
console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource'
--header "Authorization: Bearer $ACCESS_TOKEN"
--header 'Content-Type: application/json'
--data-raw
'{
  "name": "idp_saml_isv",
  "description": "This is a saml isv config",
  "protocol": "saml",
  "type": "isv",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "userID",
      "given_name": "given_name",
      "family_name": "family_name",
      "groups": "groupIds",
      "email": "email"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  },
  "scim_config": {
    "scim_base_path": "https://bedrock-iam.verify.ibm.com/v2.0/",
    "grant_type": "client_credentials",
    "token_url": "https://bedrock-iam.verify.ibm.com/v1.0/endpoint/default/token",
    "client_id": "Your_isv_client_id",
    "client_secret": "Your_isv_client_secret",
    "scim_attribute_mappings": {
      "user": {
        "principalName": "userName",
        "givenName": "name.givenName",
        "middleName": "name.middleName",
        "familyName": "name.familyName",
        "formatted": "name.formatted"
      },
      "group": {
        "principalName": "displayName",
        "created": "meta.created",
        "lastModified": "meta.lastModified"
      }
    }
  },
  "jit": "false"
}'

```

Sample response - registration with IBM Security Verify

```

{
  "status": "success",
  "message": "Identity provider {idp_saml_isv} is successfully registered with
unique identifier defaultSP"
}

```

Sample error message - registration with IBM Security Verify

```

401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema error:schema validation error followed by error"
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
200 OK Response

```

IdP V3 registration with Okta

The following example shows IdP V3 registration with Okta.

API details - registration with Okta - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/

Command

POST

Command output format

application/json

Table 1. Command parameters

Mapping	Description
<code>token_attribute_mappings</code>	A custom mapping that you can provide based on the SAML attribute mapping that is configured at Okta. You must modify the values, not the key, based on the Okta SAML attribute configuration. For more information about attributes, see step 9 in Configure CloudPak as SAML service provider (SP) at OKTA and Okta doc for SAML attributes .
<code>scim_attribute_mappings</code>	Provide the mapping attributes as shown in the curl command for OKTA registration as SAML IdP . The defined mapping is used to filter query, and to create attributes and responses to be sent.

Mapping	Description
<code>redirect_url</code>	Only applicable for the IdP "type": "okta". Must be an array. The <code>redirect_url</code> is used to create the OIDC client registration for the OKTA instance. For a non-IBM OIN app, the <code>redirect_uri</code> can be one of the following URIs based on the environment (a standard org, an EMEA org, or an Okta Preview org): https://system-admin.okta.com/admin/app/cpc/\${appName}/oauth/callback https://system-admin.okta-emea.com/admin/app/cpc/\${appName}/oauth/callback https://system-admin.oktapreview.com/admin/app/cpc/\${appName}/oauth/callback Note: Replace the <code>\${appName}</code> with the variable name for your application instance. If you don't know the variable name for application instance, you can find it as part of the App embed link from the browser by opening the embed link, or you can also check the Name attribute for the application at the <code>/Apps API</code> endpoint. For example, if the URL embedded link is https://dev-68798908-admin.okta.com/admin/app/dev-68798908_iamsaml_3/instance/00a6vnyhkm86B4xha5d7/#tab-general, the <code>\${appName}</code> is <code>dev-68798908_iamsaml_3</code> and so the <code>redirect_uri</code> is https://system-admin.okta.com/admin/app/cpc/dev-68798908_iamsaml_3/oauth/callback. For more information, see Okta documentation.

Sample curl command - registration with Okta

```
curl --location --request POST \
'https://cpconsoleroute/idprovider/v3/auth/idsource' \
--header 'Authorization: Bearer $accessToken' \
--header 'Content-Type: application/json' \
--data-raw '{
  "name": "OKTASAML",
  "description": "OKTASAML",
  "protocol": "saml",
  "type": "okta",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "uid",
      "given_name": "firstName",
      "family_name": "lastName",
      "groups": "groups",
      "email": "emailAddress",
      "first_name": "firstName",
      "last_name": "lastName"
    },
    "idp_metadata": "Base64-encoded-metadata"
  },
  "jit": false,
  "scim_config": {
    "redirect_url": ["okta_redirect_url"],
    "scim_attribute_mappings": {
      "user": {
        "principalName": "userName",
        "name": {
          "givenName": "givenName",
          "familyName": "familyName"
        },
        "displayName": "displayName",
        "emails": [
          {
            "value": "emails",
            "type": "home"
          }
        ]
      }
    }
  }
}
```

```

    },
    ],
    "id": "id",
    "userName": "userName"
  },
  "group": {
    "principalName": "displayName",
    "id": "displayName"
  }
}
}
}'

```

Sample response - registration with Okta

```

{
  "status": "success",
  "message": "Identity provider {oktasaml} is successfully registered with
unique identifier : defaultSP, oktaclient created with clientId:
yx3a5pwtdrmn0sf8ockyf97dg55lswo7 , client secret: okta-oidcclient-secret"
}

```

Sample error message - registration with Okta

```

401 "error": "Insufficient user permission for role : rolenamename"
400 "error": "schema error:schema validation error followed by error"
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
200 OK Response

```

SAML registration without any dependency

API details - registration without any dependency - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/

Command

Command output format

application/json

See the following notes:

- You can find the SAML attribute and claim names from the IdP administration. Specify the SAML attributes and claim names as the values in the **token_attribute_mappings** attributes.
- You can also provide an optional attribute named **uniqueSecurityName** in **token_attribute_mappings** payload to map the IdP attributes. By default, the **uniqueSecurityName** attribute is mapped to the **Subject** in the response of **userinfo** or **introspect** endpoint.

Sample curl command for SAML registration with w3id SAML provider

```
curl -k -X POST 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "w3id-sample-saml",
  "description": "w3id-sample-saml-test",
  "protocol": "saml",
  "type": "default",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "uid",
      "given_name": "firstName",
      "family_name": "lastName",
      "groups": "blueGroups",
      "email": "emailAddress"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  },
  "jit": true
}'
```

Sample response - registration with w3id SAML provider

```
{
  "status": "success",
  "message": "Identity provider {w3id-sample-saml} is successfully registered with unique identifier defaultSP"
}
```

Sample error message - registration with w3id SAML provider

```
401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema error:schema validation error followed by error"
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
200 OK Response
```

SAML with LDAP dependency registration

The following example is provided for IBM W3 SAML provider and with dependency of an IBM Tivoli Directory Server LDAP. The `token_attribute_mappings` might change based on the SAML provider.

API details - SAML with LDAP dependency registration - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/

Command

POST

Command output format

application/json

Note: When LDAP is configured, the response for `userinfo` or `introspect` endpoints are directly collected from liberty. Therefore, the attributes that are provided in the `token_attribute_mappings` payload are not available in the response of the `userinfo` or `introspect` endpoints.

Sample curl command - SAML with LDAP dependency registration

```
curl -k -X POST 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "saml-test",
  "description": "saml bluepages with ldap",
  "protocol": "saml",
  "type": "default",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "uid",
      "given_name": "firstName",
      "family_name": "lastName",
      "groups": "blueGroups",
      "email": "email"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  }
}
```

```

    },
    "ldap_config": {
      "ldap_id": "<existing_ldap_id>"
    },
    "jit": false
  }
}'

```

Sample response - SAML with LDAP dependency registration

```

{
  "status": "success",
  "message": "Identity provider {saml-test} is successfully registered with unique identifier defaultSP"
}

```

Sample error message - SAML with LDAP dependency registration

```

401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema error:schema validation error followed by error"
400 "error": "duplicate : Idp with protocol=saml is already created"
500 "error": "Other errors"
200 OK Response

```

Getting SAML registration by UID

You can get the IdP registration by the following ways:

- By name
- By list
- By query

Getting IdP V3 registration by name

The following example shows how to get the details of IdP V3 registration by name:

API details - registration by UID - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/defaultSP

Command

GET

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Sample curl command - registration by UID

```
curl -k -X GET 'https://cp-console.apps.mycluster.mydomain.com/idprovider/v3/auth/idsource/defaultSP' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - registration by UID

```
{
  "name": "w3id-sample-saml",
  "description": "w3id-sample-saml-test",
  "protocol": "saml",
  "type": "default",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "uid",
      "given_name": "firstName",
      "family_name": "lastName",
      "groups": "blueGroups",
      "email": "emailAddress"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  },
  "jit": true,
  "uid": "defaultSP"
}
```

Sample error message - registration by UID

```
401 "error": "Insufficient user permission for role : rolename"
404 "error": "Cannot find {defaultSP}"
500 "error": "Other errors"
200 OK Response
```

Getting the list of IdP V3 registration

The following example shows how to get the list of IdP V3 registration:

Note: This API returns all registered IdPs of SAML or OIDC clients through IdP V3 API.

API details - list of IdP V3 registration - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource

Command

GET

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Sample curl command - IdP V3 registration

```
curl -k -X GET 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - IdP V3 registration

```
{
  "idp": [
    {
      "name": "w3id-sample-saml",
      "description": "w3id-sample-saml-test",
      "protocol": "saml",
      "type": "default",
      "idp_config": {
        "token_attribute_mappings": {
          "sub": "uid",
          "given_name": "firstName",
          "family_name": "lastName",
          "groups": "blueGroups",
          "email": "emailAddress"
        },
        "idp_metadata": "<base64-encoded-idp-xml>"
      },
      "jit": true,
      "uid": "defaultSP"
    }
  ]
}
```

Sample error message - IdP V3 registration

```
401 "error": "Insufficient user permission for role : rolename"
500 "error": "Other errors"
200 OK Response
```

Getting SAML registration by query

The following example shows how to get the IdP V3 registration by query:

API details - SAML registration by query - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

GET

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Sample curl command - SAML registration by query

```
curl -k -X GET 'https://cp-
console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource?
protocol=saml '
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - SAML registration by query

```
{
  "idp": [
    {
```

```

    "name": "w3id-sample-saml",
    "description": "w3id-sample-saml-test",
    "protocol": "saml",
    "type": "default",
    "idp_config": {
      "token_attribute_mappings": {
        "sub": "uid",
        "given_name": "firstName",
        "family_name": "lastName",
        "groups": "blueGroups",
        "email": "emailAddress"
      },
      "idp_metadata": "<base64-encoded-idp-xml>"
    },
    "jit": true,
    "uid": "defaultSP"
  }
]
}

```

Sample error message - SAML registration by query

```

401 "error": "Insufficient user permission for role : rolename"
500 "error": "Other errors"
200 OK Response

```

Updating SAML registration

To update the SAML registration, you require the IdP UID. To get the IdP UID, see [Get the list of IdP V3 registration](#). Alternatively, you can use the UID as `defaultSP` because only one SAML registration at an instance is supported for now.

The following example shows how to update the IdP registration:

API details - Update SAML registration - PUT

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

PUT

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Sample curl command - Update SAML registration

```
curl -k -X PUT 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/defaultSP' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer $ACCESS_TOKEN' \
--data-raw '{
  "name": "w3id-sample-saml",
  "description": "this is plain saml testing",
  "protocol": "saml",
  "type": "default",
  "idp_config": {
    "token_attribute_mappings": {
      "sub": "uid",
      "given_name": "firstName",
      "family_name": "lastName",
      "groups": "blueGroups",
      "email": "emailAddress"
    },
    "idp_metadata": "<base64-encoded-idp-xml>"
  },
  "jit": true
}'
```

Sample response - Update SAML registration

```
{
  "status": "success",
  "message": "{defaultSP} is Updated."
}
```

Sample error message - Update SAML registration

```
401 "error": "Insufficient user permission for role : rolename"
400 "error": "schema validation error:followed by error"
500 "error": "Other errors"
200 OK Response
```

Deleting SAML registration

To delete the SAML registration, you require the IdP UID. To get the IdP UID, see [Get the list of IdP V3 registration](#). Alternatively, you can use the UID as `defaultSP` because only one SAML registration at an instance is supported for now.

The following example shows how to delete the IdP registration:

API details - Delete SAML registration - DELETE

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

DELETE

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#).

Sample curl command - Delete SAML registration

```
curl -k -X DELETE 'https://cp-  
console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/defaultSP  
' \  
--header 'Content-Type: application/json' \  
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - Delete SAML registration

```
{  
  "status": "success",  
  "message": "{defaultSP} is deleted"  
}
```

Sample error message - Delete SAML registration

```
400 "error": "Insufficient user permission for role : rolename"  
404 "error": "Document not found"  
500 "error": "Other errors"  
202 OK Response
```

SAML metadata download by using IdP V3

You can export the SAML metadata by using IdP V3 API.

Details of downloading SAML metadata by using IdP V3 API.

Downloading the SAML metadata by using IdP V3 API

The following example shows how to export SAML metadata by using the `samlmetadata` API:

API details - download SAML metadata - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

/idprovider/v3/auth/idsource/

Command

GET

Command output format

application/xml

Sample curl command - download SAML metadata

```
curl -k -v -X GET 'https://<cluster_address>/idprovider/v3/auth/idsource/<SERVICE_PROVIDER_ID>' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Example command - download SAML metadata

```
curl -k -v -X GET 'https://cp-console.apps.cp.fyre.ibm.com/idprovider/v3/auth/idsource/defaultSP' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Note: If you choose to download SAML metadata API, the response code shows 200 OK status with SAML metadata sample.

Sample response - download SAML metadata

```

<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://cp-console.apps.tamil-bedrock-
dev.cp.fyre.ibm.com/ibm/saml20/defaultSP">
  <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
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>MIIDWjCCAkKgAwIBAgIRALy+U3ooWL4WotnPukuixTswDQYJKoZIhvcNAQELBQ
          AWHDEaMBGGA1UE
          AxMRY3MtY2EtY2VydGhmaWNhdGUwHhcNMjIwNDE4MTIyODI4WhcNMjMwNDE4MTIyODI4WjAdMRsw
          GQYDVQQDExJtYW5hZ2VtZW50LWluZ3Jlc3MwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
          AQCh0rYALFRA104/DGWLlfiuftuoxT+Ab2FzsnmkO1q8Iu/SUhLf5FFnb05BrcemrL+00JcZEyR
          ALdxliWSWPG/I/9Uj+BcVlVdQupND/RQTQHS5ECYEoLJJvHkFHVj7+/WxyN3eYmq1z9OZHkP17f1
          t79HDOYPZSKVzhLhZ9MsLM/G3xIF8feRpJR0QSYUUKHB6gDXZ4Kbfui7kd/LMRQK820psA8z/Bjy
          dYQ2SXXUo4BfgpUxjN+VMldGUy0khkuUSdNklW9pLmZdT1FH7FjWH9zBWOOhz8szr+Pr4ZavHmFL
          SUP4nDMpFeH/v6+cBm48LrezWAAUzZuSTYxRQeuhAgMBAAGjgZUwgZiWdGyDVR0PAQH/BAQDAgWg
          MBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUzW7sLnme6JsG
          DZM/SC2ZRuWS/OswPAYDVR0RBDUwM4IxY3AtY29uc29sZS5hcHBzLnRhbWlsLWJlZlZlZjY2stZGV2
          LmNwLmZ5cmUuaWJtLmNvbTANBgkqhkiG9w0BAQsFAAOCAQEASHgDST++rAi79e0P5TXrUF2QQ7dc
          o2pgiv/xNKvwm/URzaSGZ1CYq69hXOd+SFZ0sverFFEQZhBXxKVIDGFMT2Jl1189sTtQsQnodbpj
          gwQxB7c4KbqI4EKU33jj9lT5CmmzHWtO5cPa7br4nOcPKp7b12q79/aDbP7GYLmp8sGMacopq5J7
          vAkL3O3De3sv2p7wg5nHB0JBA/k7Ecd/34gKpftfwZxNsBAXL+dfwOdHjK1bSlHoyo5F7gjwvLVx
          dCGdw0AfsNuW0efyGEK2+bA9bc+pzGQf6vKaaSQpdL/YC6b5QLpQkpculw1+lLESFJkdEzmM2VRB
          XfgckB39kg==</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>MIIDWjCCAkKgAwIBAgIRALy+U3ooWL4WotnPukuixTswDQYJKoZIhvcNAQELBQ
          AWHDEaMBGGA1UE
          AxMRY3MtY2EtY2VydGhmaWNhdGUwHhcNMjIwNDE4MTIyODI4WhcNMjMwNDE4MTIyODI4WjAdMRsw
          GQYDVQQDExJtYW5hZ2VtZW50LWluZ3Jlc3MwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB
          AQCh0rYALFRA104/DGWLlfiuftuoxT+Ab2FzsnmkO1q8Iu/SUhLf5FFnb05BrcemrL+00JcZEyR
          ALdxliWSWPG/I/9Uj+BcVlVdQupND/RQTQHS5ECYEoLJJvHkFHVj7+/WxyN3eYmq1z9OZHkP17f1
          t79HDOYPZSKVzhLhZ9MsLM/G3xIF8feRpJR0QSYUUKHB6gDXZ4Kbfui7kd/LMRQK820psA8z/Bjy
          dYQ2SXXUo4BfgpUxjN+VMldGUy0khkuUSdNklW9pLmZdT1FH7FjWH9zBWOOhz8szr+Pr4ZavHmFL
          SUP4nDMpFeH/v6+cBm48LrezWAAUzZuSTYxRQeuhAgMBAAGjgZUwgZiWdGyDVR0PAQH/BAQDAgWg
          MBMGA1UdJQQMMAoGCCsGAQUFBwMBMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUzW7sLnme6JsG
          DZM/SC2ZRuWS/OswPAYDVR0RBDUwM4IxY3AtY29uc29sZS5hcHBzLnRhbWlsLWJlZlZlZjY2stZGV2
          LmNwLmZ5cmUuaWJtLmNvbTANBgkqhkiG9w0BAQsFAAOCAQEASHgDST++rAi79e0P5TXrUF2QQ7dc
          o2pgiv/xNKvwm/URzaSGZ1CYq69hXOd+SFZ0sverFFEQZhBXxKVIDGFMT2Jl1189sTtQsQnodbpj
          gwQxB7c4KbqI4EKU33jj9lT5CmmzHWtO5cPa7br4nOcPKp7b12q79/aDbP7GYLmp8sGMacopq5J7
          vAkL3O3De3sv2p7wg5nHB0JBA/k7Ecd/34gKpftfwZxNsBAXL+dfwOdHjK1bSlHoyo5F7gjwvLVx
          dCGdw0AfsNuW0efyGEK2+bA9bc+pzGQf6vKaaSQpdL/YC6b5QLpQkpculw1+lLESFJkdEzmM2VRB
          XfgckB39kg==</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://cp-
console.apps.tamil-bedrock-dev.cp.fyre.ibm.com/ibm/saml20/defaultSP/slo"/>
    <md:AssertionConsumerService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://cp-
console.apps.tamil-bedrock-dev.cp.fyre.ibm.com/ibm/saml20/defaultSP/acs" index="0"
isDefault="true"/>
  </md:SPSSODescriptor>
</md:EntityDescriptor>

```

Configuring LDAP connection by using IdP V3 API

IMPORTANT: Creating and deleting an LDAP connection simultaneously can cause misconfiguration at the backend. Therefore, avoid creating and deleting the LDAP connection simultaneously.

Before you begin: Ensure that you are informed about the other LDAP configuration parameters that are used in the Identity Provider V3 APIs. For more information, see [LDAP authentication](#), [LDAP filters](#) and [Default LDAP filters by LDAP type](#). The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).

Validating LDAP configuration while creating the LDAP connection using IdP V3 API

The following example shows how to validate the LDAP connection while creating an LDAP connection by using IdP V3 API:

API details - validate LDAP connection while creating - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/ldap/validateConnection

Command

POST

Command output format

application/json

See the following notes:

- If you are setting up multiple LDAP connections, you must individually connect to each directory.
- In the curl command, you must use a base64-encoded password in the "**LDAP_BINDPASSWORD**" parameter. To encode the password, use the following command:

```
echo -n <password> | base64
```

Following is an example output:

UGFzc3cwcmQ=

Sample curl command for validating LDAP configuration while creating an LDAP connection

```
curl -k -X POST 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/ldap/validateConnection' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer $ACCESS_TOKEN'
--data-raw
'{
  "name": "openLDAP",
  "idp_config": {
    "ldap_url": "ldap://9.35.210.15:389",
    "ldap_basedn": "dc=ibm,dc=com",
    "ldap_binddn": "cn=admin,dc=ibm,dc=com",
    "ldap_bindpassword": "HNJat5opcqR="
  }
}'
```

Sample response - validate LDAP connection while creating

```
{"status": "success", "result": "Successfully validated LDAP configuration."}%
```

Validating LDAP configuration while updating the LDAP connection by using IdP V3 API

The following example shows how to validate the LDAP connection while you update an LDAP connection by using IdP V3 API:

API details - validate LDAP connection while updating - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/ldap/validateConnection

Command

POST

Command output format

application/json

See the following notes:

- The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).
- In the curl command, you must use a base64-encoded password in the "LDAP_BINDPASSWORD" parameter. To encode the password, use the following command:

```
echo -n <password> | base64
```

Example output:

```
UGFzc3cwcmQ=
```

Sample curl command for validating LDAP configuration while you update an LDAP connection

```
curl -k -X POST 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/ldap/validateConnection' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer $ACCESS_TOKEN'
--data-raw
'{
  "name": "openLDAP",
  "idp_config": {
    "ldap_url": "ldap://9.35.210.15:389",
    "ldap_basedn": "dc=ibm,dc=com",
    "ldap_binddn": "cn=admin,dc=ibm,dc=com"
  }
}'
```

Sample response - validate LDAP connection while updating

```
{"status": "success", "result": "Successfully validated LDAP configuration."}%
```

Registering an LDAP connection by using IdP V3 API

See the following notes:

- The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).
- In the curl command, you must use a base64-encoded password in the "LDAP_BINDPASSWORD" parameter. To encode the password, use the following command:

```
echo -n <password> | base64
```

Example output:

```
UGFzc3cwcmQ=
```

The following example shows how to register an LDAP connection by using IdP V3 API:

API details - register an LDAP connection - POST

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/

Command

POST

Command output format

application/json

The following sample curl command is for creating a custom LDAP connection. Replace the "**type**" parameter value with your LDAP type. For more information about the supported LDAP types, see [Configuring LDAP connection](#).

You must also replace the parameter values in the "**idp_config**" section based on your LDAP type. For more information about the parameter values for each LDAP type, see [Default LDAP filters by LDAP type](#).

Sample curl command for creating a custom LDAP connection

```
curl -k -X POST 'https://cp-
console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "openldap",
  "description": "",
  "protocol": "ldap",
  "type": "Custom",
  "idp_config": {
    "ldap_id": "openldap",
    "ldap_realm": "REALM",
    "ldap_url": "ldap://9.30.253.13:389",
    "ldap_basedn": "dc=ibm,dc=com",
    "ldap_binddn": "cn=admin,dc=ibm,dc=com",
    "ldap_bindpassword": "UGFzc3cwcmQ=",
    "ldap_type": "Custom",
    "ldap_ignorecase": "false",
    "ldap_userfilter": "(&(uid=%v)(objectclass=person))",
    "ldap_useridmap": "*:uid",
    "ldap_groupfilter": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "ldap_groupidmap": "*:cn",
    "ldap_groupmemberidmap": "groupOfUniqueNames:uniqueMember",
    "ldap_nestedsearch": "false",
    "ldap_pagingsearch": "false"
  }
}'
```

Getting information about all LDAP connections

The following example shows how to get the list of all the registered LDAP directories:

API details - information about all LDAP connections - GET

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource?protocol=ldap

Command

GET

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).

Sample curl command to get the list of registered LDAP connections

```
curl -k -X GET 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/?protocol=ldap' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - information about all LDAP connections

```
{ "id": "8967e54c-7f6f-4755-a944-cf4d2a4b8a64", "LDAP_ID": "openldap", "LDAP_REALM": "REALM", "LDAP_HOST": "corp.abc.com", "LDAP_PORT": "389", "LDAP_IGNORECASE": "false", "LDAP_BASEDN": "dc=ibm,dc=com", "LDAP_BINDDN": "cn=admin,dc=ibm,dc=com", "LDAP_TYPE": "Custom", "LDAP_USERFILTER": "(&(uid=%v)(objectclass=person))", "LDAP_GROUPFILTER": "(&(cn=%v)(objectclass=groupOfUniqueNames))", "LDAP_USERIDMAP": "*:uid", "LDAP_GROUPIDMAP": "*:cn", "LDAP_GROUPMEMBERIDMAP": "groupOfUniqueNames:uniquemember", "LDAP_URL": "ldap://corp.abc.com:389", "LDAP_PROTOCOL": "ldap" }
```

Getting information about LDAP connection by using UID

The following example shows how to get the registered LDAP directory by UID:

API details - LDAP information by UID - GET

API version

1.0.0

API URI components**Scheme**

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

GET

Command output format

application/json

Note: The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).

Sample curl command to get the list of registered LDAP connections by UID

```
curl -k -X GET 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/:UID' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - LDAP information by UID

```
{"id":"8967e54c-7f6f-4755-a944-cf4d2a4b8a64","LDAP_ID":"openldap","LDAP_REALM":"REALM","LDAP_HOST":"corp.abc.com","LDAP_PORT":"389","LDAP_IGNORECASE":"false","LDAP_BASEDN":"dc=ibm,dc=com","LDAP_BINDDN":"cn=admin,dc=ibm,dc=com","LDAP_TYPE":"Custom","LDAP_USERFILTER":"(&(uid=%v)(objectclass=person))","LDAP_GROUPFILTER":"(&(cn=%v)(objectclass=groupOfUniqueNames))","LDAP_USERIDMAP":"*:uid","LDAP_GROUPIDMAP":"*:cn","LDAP_GROUPMEMBERIDMAP":"groupOfUniqueNames:uniquemember","LDAP_URL":"ldap://corp.abc.com:389","LDAP_PROTOCOL":"ldap"}
```

Updating an LDAP connection by using IdP V3 API

See the following notes:

- Use **UID** to update the LDAP connection. To get the UID, see [Getting information about LDAP connection by using UID](#).
- The curl command contains different schema elements. To understand the use of these schema elements, see [Different schema elements for IdP V3](#) and [Schema elements for LDAP connection](#).

- In the curl command, you must use a base64-encoded password in the "LDAP_BINDPASSWORD" parameter. To encode the password, use the following command:

```
echo -n <password> | base64
```

The following is an example output:

```
UGFzc3cwcmQ=
```

The following example shows how to update an LDAP connection by using IdP V3 API:

API details - update an LDAP connection - PUT

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

PUT

Command output format

application/json

Sample curl command to update an LDAP connection

```
curl -k -X PUT 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/8967e54c-7f6f-4755-a944-cf4d2a4b8a64' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw
'{
  "name": "openldap",
  "description": "",
  "protocol": "ldap",
  "type": "Custom",
  "idp_config": {
    "ldap_id": "openldap",
    "ldap_realm": "REALM",
    "ldap_url": "ldap://9.35.210.15:389",
    "ldap_host": "9.35.210.15",
    "ldap_port": "389",
    "ldap_protocol": "ldap",
    "ldap_basedn": "dc=ibm,dc=com",
```

```

    "ldap_binddn": "cn=admin,dc=ibm,dc=com",
    "ldap_bindpassword": "UGFzc3cwcmQ=",
    "ldap_type": "Custom",
    "ldap_ignorecase": "false",
    "ldap_userfilter": "(&(uid=%v)(objectclass=person))",
    "ldap_useridmap": "*:uid",
    "ldap_groupfilter": "(&(cn=%v)(objectclass=groupOfUniqueNames))",
    "ldap_groupidmap": "*:cn",
    "ldap_groupmemberidmap": "groupOfUniqueNames:uniqueMember",
    "ldap_nestedsearch": "false",
    "ldap_pagingsearch": "false"
  }
}'

```

Sample response - update an LDAP connection

```

{"status": "success", "message": "{8967e54c-7f6f-4755-a944-cf4d2a4b8a64} is Updated."}

```

Deleting an LDAP connection by using IdP V3 API

The following example shows how to delete an LDAP connection by using IdP V3 API:

API details - delete an LDAP connection - DELETE

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/:UID

Command

DELETE

Command output format

application/json

Note: To get the UID, see [Getting information about LDAP connection by using UID](#).

Sample curl command to delete an LDAP connection

```
curl -k -X DELETE 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/8967e54c-7f6f-4755-a944-cf4d2a4b8a64' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
```

Sample response - delete an LDAP connection

```
{"status": "success", "message": "{8967e54c-7f6f-4755-a944-cf4d2a4b8a64} is deleted"}
```

Disabling OpenShift authentication by using IdP V3 API

The following example shows how to disable Red Hat OpenShift authentication by using IdP V3 API:

API details - disable OpenShift authentication - PUT

API version

1.0.0

API URI components

Scheme

HTTPS

Host IP

Cluster address

Port number

Cluster port

Path

idprovider/v3/auth/idsource/osauth

Command

PUT

Command output format

application/json

Sample curl command to disable Red Hat OpenShift authentication

```
curl -k -X PUT 'https://cp-console.apps.mycluster.mydomain.fyre.ibm.com/idprovider/v3/auth/idsource/osauth' \
--header 'Content-Type: application/json' \
--header "Authorization: Bearer $ACCESS_TOKEN"
--data-raw '{
  "uid": "osauth",
  "description": "OpenShift Authentication enablement",
  "enabled": "false",
  "idp_config": {
    "ROKS_URL": "https://oauth-openshift.apps.cp3custom.cp.fyre.ibm.com",
```



```

    "ROKS_USER_PREFIX": ""
  },
  "name": "OpenShift",
  "protocol": "oauth",
  "type": "OpenShiftAuthentication"
}'

```

Sample response - disable OpenShift authentication

```

{
  "status": "success",
  "message": "{osauth} is Updated."
}

```

Getting the host details

1. Install OC client and connect to the Red Hat® OpenShift® Container Platform cluster. By default, Red Hat® OpenShift® Container Platform is enabled automatically once you install foundational services.
2. Get the hostname from the route available in the namespace where the foundational services is installed.

```

oc get route -n <your-foundational services-namespace> cp-console -o
jsonpath='{.spec.host}' && echo

```

Sample output:

```

cp-console.apps.mycluster.mydomain.com

```

Different schema elements for IdP V3 APIs

See the following mandatory fields and their description:

- **name**: Name of the IdP that you want to register.
- **description**: Short description of the registration.
- **protocol**: Type of protocol. Currently, OIDC, SAML, and LDAP are the only supported protocols.
- **type**: Type of configured LDAP, OIDC, or SAML IdP. For example, Google, Microsoft, IBM ISV, and Okta for OIDC; **isv** and **default** for SAML.
- **idp_config**: Contains the following subfields:
 - OIDC has the following fields:
 - **discovery_url**: A well-known endpoint by OIDC provider.
 - **client_id**: A client ID that is provided by the registered application in the IdP.
 - **client_secret**: A client secret that is provided by the registered application in the IdP.
 - SAML has the following fields:
 - **idp_metadata**: A base64-encoded value of IdP metadata XML file. The maximum accepted size of the metadata file is **2 MB**. If the size of the metadata file is more than 2 MB, an error message is displayed during the SAML registration.
 - **token_attribute_mappings**: Used to map the token attributes with the IdP user attributes. You need to replace only the **values_to_be_mapped** based on your IDP configuration at the backend.
 - Sample mapping:

```

"token_attribute_mappings": {
  "groups": "values_to_be_mapped",
  "given_name": "values_to_be_mapped",
  "family_name": "values_to_be_mapped",
  "sub": "values_to_be_mapped",
  "email": "values_to_be_mapped"
}

```

The mapped values are case-sensitive.

Schema elements for SAML with SCIM dependency

See the following schema elements:

- **scim_config**: Used to provide SCIM configuration and SCIM attribute mapping information.
 - **grant_type** - Identifies the token generation flow. The **client_credentials** value is used to issue the tokens to access the SCIM APIs.
 - **token_url** - Token endpoint of IdP. It is used to request the token.
 - **client_id** - Used for the client credentials flow.
 - **client_secret** - Used for the client credentials flow.

Example schema

```

"scim_config": {
  "scim_base_path": "https://bedrock-iam.verify.ibm.com/v2.0/",
  "grant_type": "client_credentials",
  "token_url": "https://bedrock-iam.verify.ibm.com/v1.0/endpoint/default/token",
  "client_id": "Your_isv_client_id",
  "client_secret": "Your_isv_client_secret",
  "scim_attribute_mappings": {
    "user": {
      "principalName": "userName",
      "givenName": "name.givenName",
      "middleName": "name.middleName",
      "familyName": "name.familyName",
      "formatted": "name.formatted"
    },
    "group": {
      "principalName": "displayName",
      "created": "meta.created",
      "lastModified": "meta.lastModified"
    }
  }
}

```

Schema elements for SAML with LDAP dependency

ldap_config is used to provide LDAP dependency information. See the following example:

```

"ldap_config":
{
  "ldap_id": "existing-ldap-connection-name"
}

```

Schema elements for LDAP connection

To know more about LDAP parameters that are used in Identity Provider APIs, see [LDAP authentication](#), [LDAP filters](#), and [Default LDAP filters by LDAP type](#).

- **LDAP_PAGINGSEARCH:** If the **LDAP_PAGINGSEARCH** parameter is set as "**false**", it indicates that the pagination option is disabled. Set this parameter to "**true**" to enable the pagination option.
- **LDAP_PAGING_SIZE:** It is a numeric field that defines the page size of the LDAP type. You can set the number as per your requirement. It is an optional field. If you do not provide the value to the **LDAP_PAGING_SIZE** parameter, the [default page size of the LDAP server](#) is considered as a request in the API query.

For example, when the LDAP type is Microsoft Active Directory and you do not provide the **LDAP_PAGING_SIZE** value in the API, then 1000 (default page size) value is considered as a request in the API. Also, the **LDAP_PAGING_SIZE** value must not exceed the default server-side page limit or the maximum configured value for the respective LDAP type.

Issue resolution API

The issue resolution API allows for the interaction with alerts, incidents and related artifacts.

Note: The API reference (Swagger) document for the issue resolution API is not interactive and is published for your understanding of the different APIs and schemas available. For more information about how to access these APIs, review the following commands and procedure.

Endpoint URLs

Issue resolution events API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/issue-resolution/v1/events/
```

Issue resolution alerts API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/issue-resolution/v1/alerts/
```

Issue resolution incidents API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/issue-resolution/v1/incidents
```

Authentication

The API uses Platform UI (Zen) authentication, so you either need a JWT token or ZenApiKey to be authenticated. The token or key is used to determine whether a user or service ID has access when you use the API.

Accessing the API with cURL

1. Log in to your cluster and authenticate your admin user credentials for running API commands:

1. Adjust the project (namespace) for your system:

```
NAMESPACE=cp4aiops
```

2. Get the route for loading your metric data and set it as an environment variable by running the following command:

```
ROUTE=$(oc -n $NAMESPACE get route | grep ibm-nginx-svc | awk '{print $2}')
```

3. Get the admin username and set it as an environment variable by running the following command

```
USER=$(oc -n $NAMESPACE get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d)
```

4. Get the password for your admin username and set it as an environment variable by running the following command:

```
PASS=$(oc -n $NAMESPACE get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d)
```

Note: Two methods of authentication are available:

- JWT Bearer Token

This token is temporary and lasts about 4 hours.

- Platform UI (ZenApiKey) API key

This key remains until deleted.

For more information about API authentication, see [Accessing APIs](#)

2. Set your authentication.

- If you are using the Platform UI (ZenApiKey) API key option, encode your admin username and API key with Base64 encoding:

```
ZENAPIKEY=$(echo "<username>:<API_key>" | base64 -w 0); echo $ZENAPIKEY
```

- If you are using the JWT Bearer Token option, set the token as an environment variable. The following command uses variables that are set in the previous step:

```
TOKEN=$(curl -k -X POST https://$ROUTE/icp4d-api/v1/authorize -H "Content-Type: application/json" -d '{"username": \"$USER\", \"password\": \"$PASS\"}' 2>/dev/null|awk -F' ' '{print $(NF-1)}')
```

3. Run a command to access the API. For example, the following command uses cURL to access the issue resolution alerts API to load alert data:

- By using the route and the Platform UI (ZenApiKey) API key option:

```
curl -k -v -X POST "https://${ROUTE}/aiops/api/issue-resolution/v1/alerts/" --header "Content-Type: application/json" --header "Authorization: ZenApiKey ${ZENAPIKEY}" --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --data @<FILEPATH/FILE>.json
```

Where **<FILEPATH/FILE>** is the full path to the JSON file that includes your sample alert data to load.

- By using the route and the JWT Bearer Token option:

```
curl -k -v -X POST "https://${ROUTE}/aiops/api/issue-resolution/v1/alerts/" --header "Content-Type: application/json" --header "Authorization: Bearer ${TOKEN}" --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --data @<FILEPATH/FILE>.json
```

Where <FILEPATH/FILE> is the full path to the JSON file that includes your sample alert data to load.

Enabling access to the Probable cause customization APIs

The Probable cause API ranks the alerts in an incident. This API uses a classifier that uses Natural Language Processing (NLP) to determine the meaning of an alert and calculates the topology path between alerts to create a ranking for each applicable alert.

Probable cause can be computed by the following two approaches:

- Word based approach is very useful when the environment does not have topological data. This approach works in partnership with the topological probable causes when the topological data is available. Word based approach allows the users to define keywords that are given an importance score. If a word is in the summary field, the alert is given that score (the sum of all words).
- With the severity based approach, the higher severity alert is assigned a larger score. A higher severity alert does not always mean that an alert is a root cause. Higher severity alert generally gives more context to the operator about the impact. These alerts within a large group is important so that an operator can get context quickly.

You can use the Probable cause API to customize the analytics for probable cause, and add edge configurations when new custom edge types are added to IBM Cloud Pak for AIOps.

Note: If the topological data includes a user-defined edge, then the probable cause API does not understand the relationships in the topology. You can see improved results when an edge type is added to the system.

The Probable cause customization APIs are exposed by default, but to access them you must generate the token first. The API uses the IBM Cloud Pak Platform UI authentication, so you need a JWT (JSON Web Token) to be authenticated. The token is used to determine whether a user or service ID has access when they use the API.

Probable Cause Customization APIs

1. Log in to the cluster.
2. Set ROUTE to `cpd` route.
 1. If you have an existing installation of Cloud Pak for AIOps on Linux, run the following command:


```
ROUTE=$(oc get ingress zen-ingress -n cp4aiops -o jsonpath='{.spec.rules[0].host}')
```
 2. If you have an existing installation of Cloud Pak for AIOps on OpenShift Container Platform, run the following command:

```
ROUTE=$(oc get route cpd -n cp4aiops --no-headers | awk '{print $2}')
```

3. Use the admin-user-details secret to set PASS.

```
PASS=$(oc get secret admin-user-details -o  
jsonpath='{.data.initial_admin_password}' -n cp4aiops | base64 -d)
```

4. Create the TOKEN.

```
TOKEN=$(curl -s -k -X POST https://$ROUTE/icp4d-api/v1/authorize -H 'Content-  
Type: application/json' -d '{"username": "admin","password": <PASS>}' | jq  
.token | sed 's/"//g')
```

Where **PASS** is the password obtained in the preceding step.

Example of accessing the APIs

You can now access the APIs, for example:

/customisation/edge_type

Use a GET request /customisation/edge_type to get the list of edges in the system.

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer  
${TOKEN}" https://$ROUTE/aiops/api/issue-  
resolution/mime/v1/customisation/edge_type -H "accept: application/json" -H "X-  
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255"
```

The label and edge names are the same as those added in the topological resource group. Some edges have different weight due to directorial. Overall, an **indegree** edge has a higher weight than an **outdegree** edge.

Note: If the system encounters an **unknown** edge type, then it uses the average of weight in the edge label group.

The following example shows a response(200) for the GET request:

```
{  
  "edge_type_list": [  
    {  
      "edge": "string",  
      "label": "AGGREGATION",  
      "outdegree": 0,  
      "indegree": 0  
    }  
  ]  
}
```

Use a POST request /customisation/edge_type to overwrite the complete list of edge types.

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Bearer  
${TOKEN}" https://$ROUTE/aiops/api/issue-  
resolution/mime/v1/customisation/edge_type -H "X-TenantID: cfd95b7e-3bc7-4006-  
a4a8-a73a79c71255" -d '{"edge_type_list":[{"edge": "newEdgeName", "label":  
"association", "outdegree": 20,"indegree": 5}]}'
```

The following example shows a response(200) for the POST request:

```
{"Result": "Success"}
```

After the path score is calculated, the summary is classified in one of the following 8 categories.

Table 1. Label and Weight

relationship for classification
of summary

Label	Weight
errorrate	1.4
exception	1.2
information	1
latency	2
saturation	2.2
stateChange	1.8
traffic	2.4
unknown	-1

In order to modify the label weight, use the API `/customisation/label_weight`.

`/customisation/label_weight`

Use a POST request `/customisation/label_weight` to add or update label weight. The request body is an array of label weight objects.

```
curl -k -X POST -H "Content-Type: application/json" -H "Authorization: Bearer  
${TOKEN}" https://$ROUTE/aiops/api/issue-  
resolution/mime/v1/customisation/label_weight -H "X-TenantID: cfd95b7e-3bc7-4006-  
a4a8-a73a79c71255" -d '{"label_weight_list":  
[{"label": "association", "weight": 1.2}]}'
```

The following example shows a response(200) for the POST request:

```
{"Result": "Success"}
```

Note: Before removing a label from the list, remove all of the training examples from the classifier. The classifier looks at the word in the summary and attempts to put the alert in the in the appropriate category.

`/customisation/words`

Use a GET request `/customisation/words` to get the list of words in alert summary that is used by probable cause for the specified tenant ID. If the GET request is successfull(200 response), the response includes an array of **words** for the alert.

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer  
${TOKEN}" https://$ROUTE/aiops/api/issue-resolution/mime/v1/customisation/words -H  
"accept: application/json" -H "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255"
```

The following example shows a response(200) for the GET request:

```
{  
  "words": [  
    {  
      "word": "error",  
      "caseSensitive": false,  
      "weight": 100  
    },  
    {  
      "word": "ECC",  
      "caseSensitive": true,  
      "weight": 10  
    }  
  ]  
}
```

If both words "error" and "ECC" are found in same alert summary then the weight(score) is added. For example, if **error** has a weight of 100 and **ECC** has a weight of 10, then the total word score is 110. The higher the score, the more likely that this alert is a root cause.

Use a POST request `/customisation/words` to update(replace) the words in the system. The request body is the JSON array of words for probable cause.

```
curl -k -X POST --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}" -H "Content-Type: application/json" https://$ROUTE/aiops/api/issue-resolution/mime/v1/customisation/words -H "accept: application/json" -H "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255" -d '{"words": [{"word": "error", "caseSensitive": false, "weight": 100}, {"word": "ECC", "caseSensitive": true, "weight": 10}]}'
```

The following example shows a response(200) for the POST request:

```
{"Result": "Success"}
```

`/customisation/severity`

Use a GET request `customisation/severity` to get the list of severities that is used by probable cause for the specified tenant ID. Based on the severity of the alert, the weight is added to the word score. Refer to the following table for more details:

Table 2. Severity and Weight relationship

Severity	Weight
1	10
2	20
3	30
4	40
5	50
6	60

If the word score is 110(Error and ECC from previous example) and the severity of the alert is critical(which is 6), then 60 is added to the score, resulting in a word score of 170.

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}" https://$ROUTE/aiops/api/issue-resolution/mime/v1/customisation/severity -H "accept: application/json" -H "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255"
```

The following example shows a response(200) for the GET request:

```
{
  "severities": [
    {
      "severity": 2,
      "weight": 20
    }
  ]
}
```

Use a POST request `/customisation/severity` to update(replace) the probable cause to use the array of severities. The request body is JSON array of severities which will overwrite the current array of severities.

The following example shows a response(200) for the POST request:

```
{"Result": "Success"}
```


Use a DELETE request `/customisation/severity` to delete all the severities for the specified tenant ID.

The following example shows a response(200) for the DELETE request:

```
{"Result": "Success"}
```

`/customisation/scoring/config`

Use a GET request `customisation/scoring/config` to get the scoring configuration parameters for the specified tenant ID. If the API request is successful(200 response), the response includes a list of the scoring configuration attributes that are turned on or off.

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}" https://$ROUTE/aiops/api/issue-resolution/mime/v1/customisation/scoring/config -H "accept: application/json" -H "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255"
```

The following example shows a response(200) for the GET request:

```
{
  "pathCaculationEnabled": true,
  "wordEnabled": true,
  "severityEnabled": true,
  "causeScoreEnabled": true,
  "itnmproccessing": true,
  "itnmCausesWeight": 110,
  "firstBoost" : true,
  "firstBoostWeight" : 100,
}
```

Use a POST request `customisation/scoring/config` to alter the scoring configuration parameters for the specified tenant ID.

The following example shows a response(200) for the POST request:

```
{"Result": "Success"}
```

For more information about probable cause ranking and probable cause determination, see [Probable cause ranking](#)

Next steps

For more information about probable cause, see [Probable cause in ChatOps](#).

For more information about the available APIs and related Swagger documentation, see the additional topics under [APIs](#). For example, for more on access tokens, see [Accessing APIs](#).

IBM Cloud Pak for AIOps Change Risk REST API

This documentation is for the IBM Cloud Pak for AIOps Change Risk REST API

Version: 1.0.0

Endpoint Root URL

Change Risk REST API root URL is as follows:

```
https://cpd-{CLUSTER_NAMESPACE}.apps.{CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/ai-manager/change-risk/v1
```

/change_requests/risk_assessment

POST

Description

Get the risk assessment by providing a change request ticket information.

Parameters

- `include_explanation` : true or false for turning on/off explanation. Explanation is included by default.

Responses

Code	Description
200	Response
400	Bad request
404	Not found
500	Internal server error
504	Internal processing timeout
default	unexpected error

Example

The sample curl command resembles the following code:

```
CLUSTER_URL=https://cpd-cp4waiops.apps.mycluster.mydomain.myco.com/aiops/api/ai-  
manager/change-risk/v1  
curl -X POST $CLUSTER_URL/change_requests/risk_assessment?include_explanation=true  
\  
-H "Authorization: Bearer $ACCESS_TOKEN" \  
-H 'Content-Type: application/json' \  
--data '{  
  "number": "CHG0775062",  
  "sys_domain": "crn:v1:bluemix:public::mil01::::",  
  "business_service": "Network",  
  "short_description": "is DATACENTER 2020-05-17 08:29:00 UTC",  
  "description": "description text field",  
  "justification": "none",  
  "backout_plan": ".",  
  "impact": "No",  
  "reason": "Replace server due to hardware issue.",  
  "state": "Closed",  
  "type": "softlayer",  
  "sys_created_by": "SoftLayer Jira.SA",  
  "assigned_to": "name@email.com",  
  "close_code": "successful",  
  "close_notes": "Change was completed successfully"  
}'
```

The preceding data field must validate against a change record schema. The response will look similar to the following:

```

{
  "id": "77f3f724727811eb9e81c8e0eb1474cb",
  "prediction_text": "Prior ticket data insights classify this as a high risk change.",
  "prediction_confidence": 0.979616,
  "explanation": {
    "words": {
      "test": 0.1,
      "run": 0.073,
      "pod": 0.059,
      "node": 0.056,
      "error": 0.041,
      "release": 0.036,
      "remove": 0.027,
      "backout": 0.025,
      "fail": 0.023,
      "plan": 0.017,
      "utc": 0.017,
      "system": 0.013,
      "previous": 0.012
    },
    "relevant_change_tickets": [
      {
        "number": "CHG0675192",
        "sys_domain": "crn:v1:bluemix:public::mil01::::",
        "business_service": "Network",
        "short_description": "is DATACENTER 2020-05-17 08:29:00 UTC",
        "description": "description text field",
        "justification": "none",
        "backout_plan": ".",
        "impact": "No",
        "reason": "Replace server due to hardware issue.",
        "state": "Closed",
        "type": "softlayer",
        "sys_created_by": "SoftLayer Jira.SA",
        "assigned_to": "name@email.com",
        "close_code": "successful",
        "close_notes": "Change was completed successfully"
        "induced_incidents": [
          {
            "description": "description text",
            "number": "INC2468135",
            "severity": "1 - High"
          }
        ]
      }
    ],
    "ticket_relevance_level": {
      "CHG0675192": 10.0
    },
    "ticket_similarity_level": {
      "CHG0675192": 1.0
    }
  }
}

```

/feedback/types/risk_assessment

POST

Description

Provide feedback for the corresponding change risk assessment

Responses

Code	Description
200	Response
400	Bad request
404	Not found
500	Internal server error
default	unexpected error

/feedback/types/risk_assessment_explanation

POST

Description

Provide feedback for the corresponding change risk assessment

Responses

Code	Description
200	Risk assessment response
400	Bad request
404	Not found
500	Internal server error
default	unexpected error

/healthcheck

GET

Summary

Gets the health of the server.

Responses

Code	Description
200	OK
500	Internal server error

Models

Change

Name	Type	Description	Required
number	string		Yes
sys_domain	string		No
business_service	string		No

Name	Type	Description	Required
short_description	string		No
description	string		No
justification	string		No
backout_plan	string		No
impact	string		No
reason	string		No
state	string		No
type	string		No
sys_created_by	string		No
assigned_to	string		Yes
close_code	string		No
close_notes	string		No

Incident

Name	Type	Description	Required
number	string		Yes
description	string		Yes
severity	string		No

Feedback

Name	Type	Description	Required
user_response	string		Yes

FeedbackRiskAssessment

Name	Type	Description	Required
assessment_id	string		Yes
feedback	object		Yes

FeedbackRiskAssessmentExplanation

Name	Type	Description	Required
assessment_id	string		Yes
relevant_ticket_number	string		Yes
feedback	object		Yes

RiskAssessment

Name	Type	Description	Required
number	string		No
id	string		Yes
prediction_text	string		Yes
prediction_confidence	float		Yes
change_risk_score	float		Yes
explanation	object		Yes

Explanation

Name	Type	Description	Required
words	object		Yes
relevant_tickets	[object & Change]		No
ticket_relevance_level	object		No
ticket_similarity_level	object		No

RelevantTickets

Name	Type	Description	Required
RelevantTickets	array		

ErrorContainer

Name	Type	Description	Required
errors	[object]		No
trace	string	<i>Example: "9daee671-916a-4678-850b-10b911f0236d"</i>	No

Errors

Name	Type	Description	Required
Errors	array		

Error

Name	Type	Description	Required
http_code	integer	<i>Example: 500</i>	No
code	string	<i>Example: "missing_field"</i>	No
message	string	<i>Example: "The id field is required."</i>	No
more_info	string		No

Metric REST API

The Metric API allows for the posting of metric data for anomaly detection and forecasting.

Note: The API reference (Swagger) document for the metric API is not interactive and is published for your understanding of the different APIs and schemas available. For more information about how to access these APIs, review the following commands and procedure.

Endpoint URLs

Metric Ingestion API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/app/metric-api/v1/metrics
```

Metric Retrieval API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/app/metric-api/v1/metrics
```

Metric Forecasting API

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/app/metric-api/v1/forecast
```

Authentication

The API uses Platform UI (Zen) authentication, so you either need a JWT token or ZenApiKey to be authenticated. The token or key is used to determine whether a user or service ID has access when you use the API.

Error handling

Metric Ingestion API error handling

Table. Metric ingestion API error codes

HTTP Error Code	Description	Recovery
200	Success	Success.
400	Bad Request	Invalid value in JSON.
403	Forbidden	The supplied authentication is not authorized to access '{namespace}'.
500	Internal Server Error	An error occurred processing your request.

Metric Retrieval API error handling

Table. Metric retrieval API error codes

HTTP Error Code	Description	Recovery
200	Success	Success.
400	Bad Request	Invalid parameters.
403	Forbidden	The supplied authentication is not authorized to access '{namespace}'.
500	Internal Server Error	An error occurred processing your request.

Metric Forecasting API error handling

Table. Metric forecasting API error codes

HTTP Error Code	Description	Recovery
200	Success	Success.
204	No Content	Success, no data is found for provided attributes.
400	Bad Request	Invalid value in JSON.

HTTP Error Code	Description	Recovery
401	Unauthorized	You are not authorized to make this request.
403	Forbidden	The supplied authentication is not authorized to access '{namespace}'.
500	Internal Server Error	An error occurred processing your request.

Ingest data

To inject data for metric analysis, you need to first log in to your cluster and authenticate for using APIs:

1. Log in to your cluster and authenticate your admin user credentials for running API commands:

1. Adjust the project (namespace) for your system:

```
NAMESPACE=cp4aiops
```

2. Get the route for loading your metric data and set it as an environment variable by running the following command:

```
ROUTE=$(oc -n $NAMESPACE get route | grep ibm-nginx-svc | awk '{print $2}')
```

3. Get the admin username and set it as an environment variable by running the following command

```
USER=$(oc -n $NAMESPACE get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_username}' | base64 -d)
```

4. Get the password for your admin username and set it as an environment variable by running the following command:

```
PASS=$(oc -n $NAMESPACE get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d)
```

Note: Two methods of authentication are available:

- JWT Bearer Token

This token is temporary and lasts about 4 hours.

- Platform UI (ZenApiKey) API key

This key remains until deleted.

For more information about API authentication, see [Accessing APIs](#)

2. Set your authentication.

- If you are using the Platform UI (ZenApiKey) API key option, encode your admin username and API key with Base64 encoding:

```
ZENAPIKEY=$(echo "<username>:<API_key>" | base64 -w 0) ; echo $ZENAPIKEY
```

- If you are using the JWT Bearer Token option, set the token as an environment variable. The following command uses variables that are set in the previous step:


```
TOKEN=$(curl -k -X POST https://$ROUTE/icp4d-api/v1/authorize -H
'Content-Type: application/json' -d "{\"username\":
\"$USER\", \"password\": \"$PASS\"}" 2>/dev/null|awk -F' ' '{print $(NF-
1)}')
```

3. Run a command to load the training data file. As an example, the following command uses cURL to access the API.

- By using the route and the Platform UI (ZenApiKey) API key option:

```
curl -k -v -X POST "https://${ROUTE}/aiops/api/app/metric-
api/v1/metrics" --header 'Content-Type: application/json' --header
"Authorization: ZenApiKey ${ZENAPIKEY}" --header 'X-TenantID: cfd95b7e-
3bc7-4006-a4a8-a73a79c71255' --data @<FILEPATH/FILE>.json
```

Where <FILEPATH/FILE> is the full path to the JSON file that includes your training data to load.

- By using the route and the JWT Bearer Token option:

```
curl -k -v -X POST "https://${ROUTE}/aiops/api/app/metric-
api/v1/metrics" --header 'Content-Type: application/json' --header
"Authorization: Bearer ${TOKEN}" --header 'X-TenantID: cfd95b7e-3bc7-
4006-a4a8-a73a79c71255' --data @<FILEPATH/FILE>.json
```

4. To ensure that data is injected, run a command to return data with the API. As an example, the following command uses cURL to access the API.

- By using the route and the Platform UI (ZenApiKey) API key option:

```
curl -v "https://${ROUTE}/aiops/api/app/metric-api/v1/metrics?
resource=MyApp&group=Latency&metric=calls.sum" --header "Authorization:
ZenApiKey ${ZENAPIKEY}" --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-
a73a79c71255' --insecure
```

- By using the route and the JWT Bearer Token option:

```
curl -v "https://${ROUTE}/aiops/api/app/metric-api/v1/metrics?
resource=MyApp&group=Latency&metric=calls.sum" --header "Authorization:
Bearer ${TOKEN}" --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-
a73a79c71255' --insecure
```

JSON schema for metrics data

JSON Payload:

```
{
  "groups": [
    {
      "timestamp": <timestamp>,
      "resourceID": "<resource ID>",
      "metrics": {
        "<kpi-1>": <value-1>,
        "<kpi-2>": <value-2>,
        ...
        "<kpi-n>": <value-n>
      },
      "attributes": {
        "group": "<metric namespace>",
        "node": "<top-level resource>"
      }
    }
  ]
}
```

```
    }
  ]
}
```

- **timestamp** - The epoch time in milliseconds (13 digits) or ISO-8601.
- **resourceID** - The resource that is being measured.
- **metrics** - A map of metric name to metric value.
- **attributes** - Anything that helps provide context to the user.
- **group** - Field that you can use to namespace a metric. For example, "customer1". Therefore, group helps distinguish metrics names.
- **node** - The top-level resource.

Example JSON Payload:

```
{
  "groups": [
    {
      "timestamp": "1642516320000",
      "resourceID": "MyApp",
      "metrics": {
        "calls.sum": 476.0,
        "latency.mean": 1.6722689
      },
      "attributes": {
        "group": "Latency",
        "node": "MyApp"
      }
    }
  ]
}
```

Note: The **attributes.group**, **resourceID**, and the **metrics** (for example, **metrics.access-denied**) fields, along with the **tenantID** (part of the header in the HTTP request) make up the unique identifier for a KPI. Therefore, as part of the metric anomaly processing and detection, any data points with different values for these fields are considered as different metrics. For example, consider the following two data points:

```
{
  "groups": [
    {
      "timestamp": 1634846130000,
      "resourceID": "10.10.10.2",
      "metrics": {
        "access-denied": 14977
      },
      "attributes": {
        "group": "Org1",
        "node": "firewall-1"
      }
    },
    {
      "timestamp": 1634846130000,
      "resourceID": "10.10.10.2",
      "metrics": {
        "access-denied": 379
      },
      "attributes": {
        "group": "Org2",
        "node": "firewall-1"
      }
    }
  ]
}
```

These two data points have the same **timestamp** value (1634846130000) and **resourceID** value (10.10.10.2), but have two different group attribute names (**Org1** and **Org2**). This difference causes the data points to be treated as different metrics.

The JSON metric field for correlating a metric anomaly with an element in a resource group depends on the data source for the corresponding topology data. For example, for topology data that is ingested through an Instana connection, the following matching rule is used for correlation: `${name}:::${uniqueId}`. As another example, for topology data that is ingested through a file observer job, the **resourceID** field is used as the identifier for topology correlation.

Unpegging, accumulator and counter attributes: For metrics that count upwards only from when the system starts, the analytics wants to work on the delta values rather than the raw values.

- To identify which metrics are pegs, accumulators, or counters in metric anomaly detection, set an attribute per metric group, for example:

```
"accumulators": "TotalConnections, SuccessfulLoginCount"
```

- Alternatively, you can set the attribute to **ALL** to apply to all metrics, for example:

```
"accumulators": "ALL"
```

Aggregation attributes: Average is the default aggregation type, except for accumulators, where sum is the default.

- To identify which metrics are to be aggregated with sum in metric anomaly detection, set an attribute per metric group, for example: `"sumAggregator": "newConnections, Logins"`
- Alternatively, you can set the attribute to **ALL** to apply to all metrics, for example:

```
"sumAggregator": "ALL"
```

Introduction

Runbook Automation is responsible for the management of runbooks and runbook instances. Runbooks are structured operational procedures which can be created, changed, and executed. They might include several types of automation and run manual, semi-automated, or fully-automated. Runbook instances are the execution records of Runbooks. They contain the status and details of one specific execution and can be changed to step through an execution and finish it. Additionally other administrative functions are presented in the API, including configuration management. To send API calls, create an API key in the application and find detailed API documentation inside the Swagger document.

Endpoint URLs

Runbook API

It is used to manage runbooks.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/runbooks
```

Runbook instance API

It is used to manage runbook executions.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/runbookinstances
```

Configuration API

It is used to manage Runbook Automation configuration.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/configuration
```

Tag API

It is used to retrieve runbook tags.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/tags
```

Usage Statistics API

It is used to retrieve Runbook Automation usage statistics.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-  
manager/rba/v1/statistics/runbookinstances
```

Status API

It is used to retrieve the status of the Runbook Automation deployment.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/status
```

User API

It is used to retrieve a list of all users who have a given role.

```
cpd-aiops.apps.{OC_CLUSTER_NAME}.{CLUSTER_DOMAIN_PREFIX}.  
{CLUSTER_TERMINAL_DOMAIN}/aiops/api/story-manager/rba/v1/users
```

Authentication

The API uses Zen authentication which requires a JWT token to be authenticated. The token is used to determine whether a user or service ID has access when they use the API.

Error handling

The Runbook Automation API returns one of the following HTTP responses for every API call.

HTTP Error Code	Description	Recovery
200	Success	The call was successful. See the output for the requested data.
204	No Content	Action completed successfully, no data is returned by the server.
400	Bad Request	The input was invalid. Read the error message and adjust your input before trying again.
401	Unauthorized	The request was not authorized. Make sure your API key is valid and the user and password are entered correctly.
403	Forbidden	The supplied authentication is not authorized to access the given API.
500	Internal Server Error	An internal error happened. A correlationId is provided. Include this ID in any request towards IBM support.

Application and topology APIs

IBM Cloud Pak® for AIOps offers a set of tools and APIs that you can use to predict, communicate, and resolve events before they become serious problems for your organization.

To learn more about how topologies are displayed in the UI, review the topology viewer documentation. For more information, see [Topology viewer](#).

API authentication

For more information on how to authenticate to use APIs, see [Accessing APIs](#).

Access to URL routes

For more information on how to access the URL routes for the topology Swagger documentation, see [Enabling access to URL routes](#).

Access the topology service Swagger through port forwarding

- [Accessing topology Swagger](#)

Swagger documentation

Review the following Swagger documentation to use Topology APIs for configure File Observer and Rest Observer jobs.

- [Topology File Observer API](#)
- [Topology Rest Observer API](#)

To learn more about how to use these jobs, such as for including custom data in topologies, see:

- [Configuring REST Observer jobs](#)
- [Configuring File Observer jobs](#)

Troubleshooting: Swagger UI not responding during high-volume query

When querying for resources using **GET /resources**, the Swagger UI may become unresponsive if there are in excess of a thousand results.

- **Cause:** This is a Swagger limitation.
- **Workaround:** None. Avoid high-volume Swagger queries.

API reference documentation

To learn more about using the topology APIs, including terminology and properties, see [Topology API reference](#).

Accessing Topology service Swagger UI

You can access the Topology service Swagger using port forwarding.

Before you begin

You need to be an administrator with `oc client` access in order to `get` secrets, services, and to enable port-forwarding in the ASM namespace.

About this task

A simple way to access the Topology service Swagger is to use the `oc port-forward` command. You can then access Swagger on your local browser through `localhost`. **Note:** Port forwarding will be enabled temporarily only. When there is no traffic, it will automatically be closed.

- To log on, use `oc login`
- To obtain the service name, use `oc get service`
- To enable port forwarding, use `oc port-forward service/aiops-topology-topology 8080`

Enable port forwarding

See the following steps as an example.

1. Log on using the `oc login` command.
2. Find the secret storing the credentials for Swagger using the `oc get secret | grep topology-asm-credentials` command.

System output:

```
aiops-topology-asm-credentials      Opaque
2          3h54m
```

3. Find the topology service secret using the `oc get service | grep topology-topology` command.

System output:

```
aiops-topology-topology      ClusterIP    1.2.3.4
<none>      8080/TCP,8081/TCP
3h49m
```

4. Enable port forwarding using the `oc port-forward service/aiops-topology-topology 8080` command.

System output:

```
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
```

Note: If the port is not accessed for a while, the connection is closed:

```
Forwarding from [::1]:8080 -> 8080
E0622 07:57:52.101053    25681 portforward.go:233] lost connection to pod
```

5. Now access the topology service swagger api in a browser on the same host as the one the oc commands were run on, at the following URL `https://localhost:8080/1.0/topology/swagger`
6. Authorise using the credentials from the secret obtained in step 3.

Enabling access to URL routes for topology observer APIs

To access the URL routes for the topology Swagger documentation, you need to update the IBM ASM Operator configuration.

About this task

The URL to access an observer API consists of **{HOST}/{PATH}** and the Swagger UI URL consists of **{HOST}/{PATH}/swagger**.

Procedure

Complete the following configuration to expose all observers for both **full** and **listen** jobs.

1. Run the following command to edit the ASM operator configuration:

```
oc edit asm aiops-topology
```

2. Search for the string **helmValuesASM:** within the **spec** section of the operator custom resource. Add **global.enableAllRoutes: true** to the list of values to enable the routes. The following example, shows this setting:

```
spec:
  helmValuesASM:
    global.enableAllRoutes: true
```

3. Save your changes and wait for the ASM operator to reload.
4. Use the **oc get route | grep** command to view the route details for an observer API to check whether the topology routes are enabled:

```
oc get route | grep topology
```

Output:

```
aiops-topology-topology-cp4aiops.apps.aiops.xxxxxx.xxxxxx.com /1.0/topology/
```

Examples

Example: View AppDynamics observer route details

The following example uses the AppDynamics observer:

```
oc get route | grep appdynamics
```

Output:

NAME	HOST/PORT		
PATH	SERVICES	PORT	
TERMINATION	WILDCARD		
aiops-topology-appdynamics-observer	aiops-topology-appdynamics-observer-		
aiops.apps.myibm.com	/1.0/appdynamics-observer	aiops-topology-	
appdynamics-observer	9148 reencrypt/Redirect	None	

Example: cURL command to the AppDynamic Observer (or job) endpoint

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer
${TOKEN}" https://aiops-topology-appdynamics-observer-
aiops.apps.myibm.com/1.0/appdynamics-observer/jobs --header 'X-TenantID: cfd95b7e-
3bc7-4006-a4a8-a73a79c71255'
{"_items": [], "_limit": 100}
```

Note: To obtain your access token, see [Accessing APIs](#).

Example Swagger UI URL: <https://aiops-topology-appdynamics-observer-aiops.apps.myibm.com/1.0/appdynamics-observer/swagger>

Next steps

For more information on the available API and related Swagger documentation, see the additional topics under [APIs](#).

Topology API reference

The following topics explain the common terminology, properties, types, and more for the available topology APIs.

Terminology

In addition to the following terms, ensure that you are familiar with the important terms that are defined within the [Glossary](#).

resource: A resource is a node in an interconnected topology, sometimes also referred to as a vertex, or simply a node. It can be anything in a user-specific topology that is designated as such, for example a hardware or virtual device, a location, a user, or an application.

edge: An edge is a relationship between resources, also simply referred to as the 'link' between resources. Edges have a label, which allocates them to a family of edges with specific behavior. It also governs how they are displayed in the UI, and an edgeType, which defines the relationship in real terms.

tenant: A tenant is represented by a globally unique identifier, its tenant ID. The default tenant ID is: cfd95b7e-3bc7-4006-a4a8-a73a79c71255

provider: A provider is usually a single data source within the scope of a tenant. The provider is a namespace for the resource's uniqueId. That uniqueId property is unique only within the scope of a provider.

status: Status is a property of one or more resources, and a single resource can have different types of status.

Each status can be in one of three states:

- open

- clear
- closed

The severity can be one of the following ratings:

- clear
- information
- indeterminate
- warning
- minor
- major
- critical

The status of a resource is derived from events, or if the resource is retrieved through the status service. It can also be supplied when resources are posted to the topology service.

Reference documentation

[Properties](#)

The Topology Service has two categories of properties, generic and user. Generic properties have fixed data types, while user-defined properties do not.

[Edge labels](#)

The topology service defines a family of labels for the edges it supports.

[Edge types](#)

All edges created in the Topology Service should define an edge **type**. The following section lists the edge types that are associated with each of the public-facing edge labels. If none of the default edge types suffice, you can create custom edge types.

[Entity types](#)

Use the Topology Service to group together resources of the same type using the **entityTypes** property, and identify them in the UI by their icon. Usually a resource has only a single entity type but as the property is a *Set*, it is possible for a resource to have more than one entity type. A number of pre-defined entity types are supplied. In addition, you can create more entity types as required.

[Status \(and state\)](#)

Resources have specific statuses, and in turn each status has a state of open, clear or closed.

[Timestamps](#)

Both vertex and edge graph elements can have multiple timestamps, as are documented here.

[Topology examples](#)

You can review examples that describe best-practice guidelines to provide you with practical information, such as implementation examples and code samples, that you can use to get started.

Properties

The topology API has two categories of properties, generic and user. Generic properties have fixed data types, while user-defined properties do not.

- Some properties have special semantics, as described in **Generic properties**.
- The pre-defined properties are listed in the **Pre-defined properties** table.
- You can add any property as required, as described in **User properties**.

Generic properties

Generic properties are few and constrained to a fixed data type. They can also be subdivided into those which are read/write and those which are read-only.

- **uniqueId**

The uniqueId is the string that is used to match resources from the same provider. It could be, for example, a UUID by which the provider can look up its own local data store for information about that device.

If you send the same resource with the same uniqueId and the same provider more than once, it will be treated as the same resource.

The provider is a project (namespace) for the resource's uniqueId. That uniqueId property is unique only within the scope of a provider.

- **matchTokens**

These tokens are used to store strings, which are significant to that resource, and might match it to events.

- **name**

This string does not have to be unique, and it should be short and memorable.

- **tags**

Tags can be used to search for resources in the topology view, and to group resources. For more information on configuring tag rules and tag-based templates, see the following sections:

- [Defining rules](#)
- [Group templates](#)

- **entityTypes**

These are defined as a set, though with usually only a single member, of one or more types this resource represents.

For more information on entityTypes, see the [Entity types](#) topic.

Predefined properties

Table. Predefined properties

Name	Type	Cardinality	Alias	Read-only	Indexed
age	integer	single		no	y
aliasIds	ID	set	_aliasIds	yes	n
beginTime	long	single	_startedAt	yes	y
changeTime	long	single	_modifiedAt	yes	n

Name	Type	Cardinality	Alias	Read-only	Indexed
createTime	long	single	_createdAt	yes	y
deleteTime	long	single		yes	n
description	string	single		no	y
edgeTenantId	ID	single	_edgeTenantId	yes	n
edgeType	string	single	_edgeType	yes	n
endTime	long	single	_deletedAt	yes	y
entityTypes	string	set		no	y
eventId	string	single		yes	n
eventManager	string	single		yes	n
expireTime	long	single	_expiredAt	yes	n
geolocation	GeoLocation	single		no	n
hasState	string	single		yes	n
icon	string	single		no	n
id	long	single		yes	n
keyIndexName	string	single		yes	n
label	string	single		yes	n
matchTokens	string	set		no	y
name	string	single		no	y
observedTime	long	single	_observedAt	yes	n
operation	string	single		yes	n
partOfExternal	Boolean	single		yes	n
prevBeginTime	long	single		yes	y
providerId	ID	single		yes	n
providerName	string	single		yes	n
reconciliationTokens	string	set		yes	n
referenceId	ID	single		yes	y
referenceNo	long	single		yes	n
serializedHashMap	HashMap	single		yes	n
severity	string	single		no	n
speed	long	single		no	y
statusType	string	single		yes	n
tags	string	set		no	y
tenantIds	ID	set	_tenantIds	yes	y
uniqueId	string	single		no	y
uuid	ID	single	_id	yes	y
version	string	single		no	y
vertexType	string	single		yes	y

User properties

User-defined properties are free-form, and are not constrained by any data type. You can add new user properties as needed.

You can define any custom properties, such as, `ipAddress`.

Note: All user-defined properties such as `ipAddress` are not in the generic set, and are stored as a serialized `blob` data type instead. The implication of this serialized data type storage convention is that these properties cannot be filtered, as they are incompatible with the `_filter` query parameter used in the REST API.

JSON record depicting both types of properties

The following picture depicts how the properties in a JSON record from an Observer can be split into both generic and user properties.

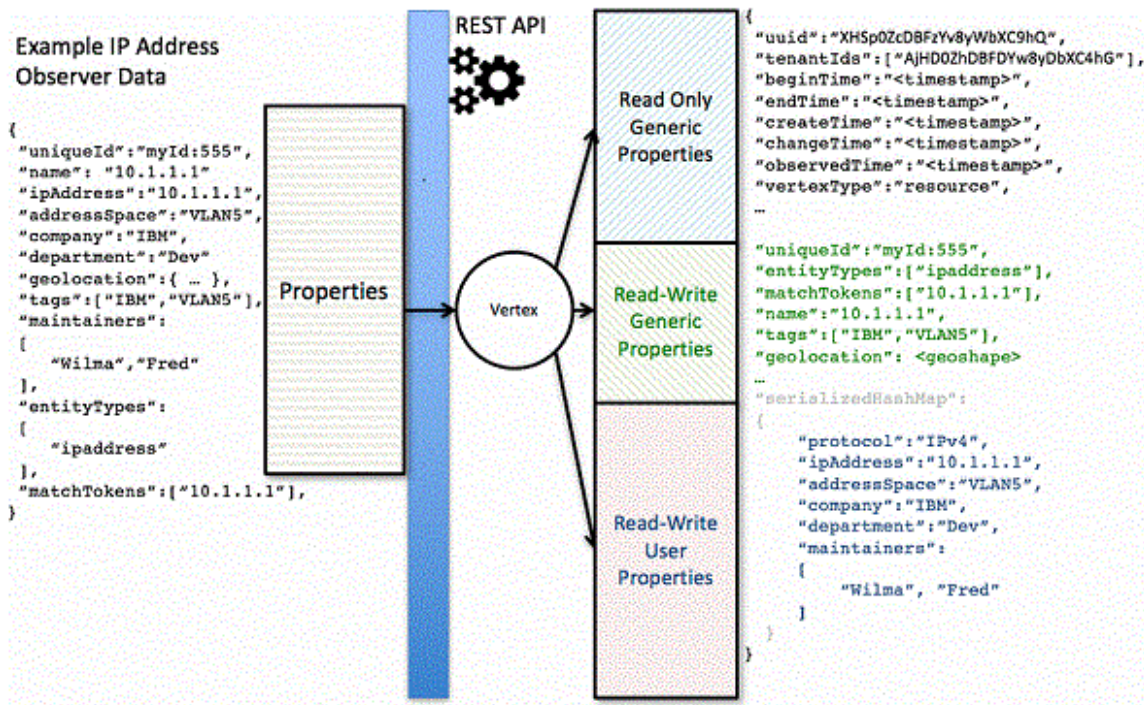


Figure. JSON record properties

Edge labels

The topology service defines a family of labels for the edges it supports.

Note: Most interactions with edges in the topology are with edge *types* rather than edge *labels*. These types can be conceptualized as instances of the edge label classes, and are documented separately. For more information, see [Edge types](#).

- **aggregation**

A relationship where the target vertex can exist independently of the source, but is needed for the source to function.

A 'parent-child' relationship where the parent (source) is aggregating the children (target).

The type of aggregation is determined by the value of the edge type.

Use this edge label to represent a resource in the UI that is composed of various elements, for example a book **contains** words.

- The direction is always from parent to child.
- Children can have multiple parents.

For more information on the types that associated with this label, see [Edge types](#).

- **association**

A 'weak' relationship where both source and target vertex can exist independently and each does not require the other to function.

The specific type of association is determined by the edge type.

Use this edge label to represent a general relationship between vertices in the UI, for example a person **has** a house.

The label itself has no direction, but a direction could be implied by the edge **type** used.

For more information on the types that associated with this label, see [Edge types](#).

- **composition**

A 'part-whole' relationship where the child (source) is a part of the parent (target).

Use when you need to create a short-cut from a node to its root, bypassing fine-grained relationships, for example a component is **partOf** a device.

- The direction is always child to parent.
- Children cannot have multiple parents.
- Composition relationships are hidden from the element-to-element view. They are used to construct the element-to-host view.
- **partOf** is the only edge type you can use for the 'composition' label, and is reserved for its exclusive use.

- **dataFlow**

A dataFlow label represents a data flow between a pair of vertices.

The specific type of data flow is qualified by properties on the edge type.

Use this label when you need to represent any form of data flow in the UI, for example a person emailing another person.

The label itself has no direction, but a direction can be implied from the edge type used.

For more information on the types that associated with this label, see [Edge types](#).

- **dependency**

A 'strong' relationship where the source depends on the target and cannot operate independently of it, and is not integral to it.

The specific type of dependency is determined by the edgeType.

Use this label when you need to represent the dependency of one resource on another in the UI, for example an application dependsOn a database.

The direction is always from the dependent resource to the independent resource.

Edge types

All edges created in the topology should have an edge **type** defined. Review the following lists of the types that are associated with each edge label. If none of the default types suffice, you can create your own custom types.

Edge **types** can be thought of as being instances of the edge **label** classes. Most interactions with edges in the topology are with *types* rather than *labels*.

Edge types for labels

Types for the aggregation label

Edge type	Description	Example
contains	The source resource is considered to contain the target resource	Slot contains a card
federates	The source resource is a federation of the target resources	Database federates nodes
members	The source resource has the target resources as its members	Subnet members are IP addresses

Types for the association label

Edge type	Description	Example
aliasOf	Depicts that one resource is an alias of another; potentially from different providers	FQDN1 is an alias of FQDN2
assignedTo	When one resource is assigned to another	The alarm is assignedTo an operator
attachedTo	When one resource is attached to another	The plug is attachedTo to the cable
classifies	When one resource is used to classify another	The government classifies the document
configures	When one resource configures or provides configuration for another resource	The file configures the application
deployedTo	Describes when one resource is deployed to another resource	The application was deployedTo the server
exposes	When one resource exposes another	The application exposes the interface
has	Generalized relationship when one resource possesses, owns, or has another resource	Host has component
implements	When one resource implements another	The class implements the interface
locatedAt	When one resource is physically located in or at another resource's location	Host is locatedAt data center

Edge type	Description	Example
manages	When one resource manages another	The boss manages the employee
monitors	When one resource monitors another	The application monitors the host
movedTo	When one resource moves to a new and different resource	The service has movedTo the host
origin	Indicates the origin of a Vertex	Device's origin is a vendor
owns	Indicates ownership of one resource by another resource	The user owns the server
rates	Can be used when one resource rates another	The manager rates the employee
resolvesTo	When one resource resolves to another	The hostname resolvesTo an address
realizes	When one resource realizes another	The hypervisor realizes the virtual machine
segregates	When one resource segregates another	The firewall segregates the network
uses	When one resource takes, employs or deploys another resource as a means of achieving something	Application uses this disk

Types for the dataFlow label

Edge type	Description	Example
accessedVia	One resource is accessed through another, typically remote.	Server is accessedVia a FQDN
bindsTo	A network layering relationship such that the source runs on top of the target.	Logical interface bindsTo a physical port
communicatesWith	A relationship whereby one resource communicates with another	The sensor communicatesWith an application
connectedTo	A relationship whereby one resource is connected to another	The laptop is connectedTo the switch
downlinkTo	One resource is down-linked to another	The controller has a downlinkTo the sensor
reachableVia	A resource is reachable through another resource	The network is reachableVia the gateway
receives	A resource receives data from another	The Mail server receives an email
routes	A relationship whereby one resource routes data for another	The device routes the data
routesVia	A relationship whereby data from one resource routes through another	The traffic routesVia the device
loadBalances	One resource which balances load for others	The load balancer loadBalances to servers
resolved	When one resource resolved something for another	DNS server resolved IP address
resolves	Represents that one resource uses another to resolve it	FQDN resolves to the address
sends	A resource sends some data to another	The application sends an SMS message

Edge type	Description	Example
traverses	Describes when one resource traverses another	The message traverses the network
uplinkTo	One resource is up-linked to another	The sensor has an uplinkTo the controller

Types for the dependency label

Edge type	Description	Example
dependsOn	A generic dependency between resources	One application dependsOn another
runsOn	A resource runs on (and therefore depends on) another resource	The service runsOn the host

Custom edge types

If no default edge types suitably represents a relationship, you can define a custom edge type through the <https://localhost:8080/1.0/topology/types/edge> POST API.

Important: A custom edge type needs to be created **before** the observation job passes any edges with the type to the topology service.

- **edgeType**

Required. The edgeType name, which must be unique and **cannot** match the edgeType name of a default edgeType, unless the edgeLabel also matches the corresponding default edge type's edgeLabel parameter.

Restriction: A scenario where both edgeType and edgeLabel match the fields of a default edge type is equivalent to manually creating a default edge type, which is not necessary as default edge types are created implicitly by the topology when needed.

- **edgeLabel**

Required. The edgeLabel of the custom edgeType, which must have one of the following labels:

- dataFlow
- dependency
- association
- aggregation

- **description**

Optional, but recommended. This should be a meaningful description of the type of relationship this edge type represents.

Example

```
curl -k -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '{
  "edgeType": "connectedTo",
  "edgeLabel": "dataFlow",
  "description": "Default relationship between two devices that exchange data"
}' 'https://localhost:8080/1.0/topology/types/edge'
```


Entity types

The Topology Service allows you to group resources of the same type that uses the **entityTypes** property, and identify them in the UI by their icon.

Usually a resource has only a single entity type, but, as the property is a *Set*, it is possible for a resource to have more than one entity type. Pre-defined associations of icons to specific entityTypes are listed in the following sections. You can create additional entity types as required.

- An entityType is defined by defining a resource with that type in the entityTypees list field.
- You can associate icons with named types in the Topology Viewer.
- You can add custom icons.

Predefined entity types

An entity type is used to map the resource vertex to an icon.

Notes:

- An administrator user can create more custom resource or entity types as required. Any observer may add new types.
- The topology service API only returns the resource or entity types in use.

To assign an entityType

1. Select a resource with the provided entityType in the Topology Viewer.
2. Select the topology tools (the **cog** icon), then navigate to the resource types.
3. Find the type to be edited, and select the icon to assign it. You can find an icon either by name from the dropdown list, or by using the **View all icons** option (the **eye** icon).

Entity type icons

The following table lists these entity types with links to their icons, if defined:

Table. Entity type icons

Entity type	Icon
aircond	
alarm	
application	
backplane	
bridge	
building	
bypass	
card	

Entity type	Icon
cell	
channel	
chassis	
circuit	
cluster	
command	
component	
computer	
container	
cpu	
credential	
daemonset	
database	
datacenter	
deployment	
device	
directory	
disk	
domain	
emailaddress	
environment	
event	
fan	
file	
firewall	
flavor	
forum	
fqdn	
gauge	
group	

Entity type	Icon
host	
hsrp	
hub	
image	
input	
ipaddress	
ipam	
loadbalancer	
location	
memory	
message	
namespace	
network	
networkaddress	
networkgear	
networkinterface	
operatingsystem	
organization	
outlet	
output	
patch	
path	
pdu	
peripherals	
person	
pod	
printer	
process	
product	

Entity type	Icon
project	
psu	
resource	
router	
routetable	
routetarget	
rsm	
rule	
sector	
securitygroup	
sensor	
server	
service	
serviceaccesspoint	
slot	
snmpsystem	
stack	
statefulset	
status	
storage	
subnet	
subscription	
switch	
swarm	
task	
tcpudpport	
template	
tenant	
tier	
tool	

Entity type	Icon
ups	
variable	f_x
version	$\begin{matrix} * \\ a \ b \end{matrix}$
vlan	VLAN
vm	
volume	
vpn	VPN
vrf	
workload	

Status (and state)

Resources have specific statuses, and in turn each status has a state of open, clear, or closed.

Status

A single status can affect multiple resources, and a single resource can have multiple different statuses. For example, a single *link down* event can generate the status for both the interface resource and the host resource, or, a single host might have *CPU* or *disk usage* status in addition to any *link down* status.

Resource status can be viewed in the Topology Viewer. An 'open' event is one with a severity other than 'clear'.

Important: When modeling resources, you must consider [Status assignment from events](#).

Tip: The Topology Service stores the event **Severity**, and nodes in the UI are colored based on severity, which always has one of the following definitions:

- clear
- indeterminate
- information
- warning
- minor
- major
- critical

Look at the severity icons in the topology viewer reference topic [Severity icons table](#).

Status assignment

The status of a single resource can be supplied, alongside other resource properties, when creating or updating a resource. Alternatively, an event can generate the status of one or more resources.

Remember: A status **always** has one of the following three states:

- **Open**

Always has a severity other than 'clear' to indicate an active issue that can require your attention.

- **Clear**

Working as expected. Also has a severity of 'clear'.

- **Closed**

No longer active or relevant, a deleted event. Also has a severity of 'clear'.

Status assignment from events

The status service receives events and tries to find matching resources in the topology service. A resource with no match tokens defined will not have events matched to it, but if found, the status of those resources is set from the event data. The default fields of the alert used to look up resource matchTokens are *resource.sourceId* and *resource.name*. The assigned status depends on the following event data:

- **matchTokens**

This property must be used to list any data that can identify (match) the resource.

Each field may be globally unique, or may be unique within the scope of a composition. In other words, a resource modeled as a composition relationship, such as **partOf**, can be distinguished from other children within the composition using these fields.

For example, either a globally unique and fully qualified domain name or IP address, or a locally unique interface name (that is, local within the scope of the host), can be used to identify the resource.

- **partOf** composition relationship

The status service uses composition relationships to match fields that are unique only within the scope of a parent.

For example, an IP address can be used to find a main node, and an interface name can be used to identify an interface within that main node.

- **Note:** If two different resources have the same **matchTokens** value, an event cannot be assigned to them unless they have been merged to become members of the same composite, **or** they have a parent-child **partOf** relationship.

Access scope

The scoping functions allows for preferential matching of IBM Cloud Pak for AIOps alerts. If similar resources and alerts in your environment exist in multiple different scopes, you can tag them with scope-specific information to allow a unique match to be made.

Such a scenario can occur (for example) under the following conditions:

- The same applications are running in test, staging and production environments.
- The same deployments are running in multiple namespaces in a single Kubernetes cluster.
- The same components are running in cloned Openstack projects.
- The same private IPs exist in multiple routing domains.

If an alert matches duplicate resources, a (case-insensitive) match of the alert scope against the resource scope is attempted.

Resource scope: Resources have an *accessScopeTokens* list property containing the following information:

- The provider of the resource, as defined by the observer job
- Optional additional scope supplied in the observer job's *Access Scope* CSV string parameter
- Scope explicitly added by some observers, such as the Kubernetes observer's *data_center* and *namespace* job parameters

Alert scope: The alert fields to use for scope are configurable.

- You can use the **AIOPS_SCOPING_FIELDS** variable defaults (*accessScope*, *scopeId*, *location*, *cluster*) to deuplicate resources when there are multiple matches on *matchTokens*.
- The values of these fields are compared with the resource property *accessScopeTokens*, with preferential matching based on case-insensitive comparison.

Troubleshooting alert lookup failures:

1. Check for a unique resource match at `GET /resources?_filter=matchTokens:{alertFieldToLookup}`
2. If the alert is older than the resource, it might not have existed when the alert processing looked it up.

Timestamps

Both vertex and edge graph elements can have multiple timestamps.

The following formats apply across all Topology APIs when specifying timestamps:

- Millisecond Epoch

Example:

```
date "+%s%3N"
1655991641000
```

- RFC-3339 / ISO-8601

Format:

```
YYYY-MM-DDThh:mm:ss.sssZ
```

Example:

```
date --universal "+%FT%T.%3NZ"
2022-06-23T13:40:41.000Z
```

- **beginTime** - The *beginTime* timestamp records the beginning of a period of time for which the resource was valid, with *endTime* marking the end of that period.

Tip: A resource can have multiple begin times in its historic record, and gaps can happen in that record if the resource was offline for periods.

- All resources and historic resources that are representations of the same thing have a distinct *beginTime*
- Resource *beginTime* together with *endTime* is used in historic graph traversals, that is, when the *_at* parameter is supplied. The period during which a resource is valid is defined as:

`atTime >= beginTime && atTime < endTime`

- A vertex or edge, which has the `beginTime` equal to the `endTime`, can be used to store audit information, such as the provider which deleted a resource. However, because it takes up zero time it does not form part of the history and is ignored by the preceding equation.
- **prevBeginTime** - If history exists for a resource, then this property is set to the `beginTime` of the most recent historical resource.
- **changeTime** - The `changeTime` timestamp records when the properties of the element last changed. The value can be less than the `observedTime`, which is updated on a POST or PUT even if no property values have changed.
- **createTime** - The `createTime` timestamp records when the element was first created in the topology service.
 - Historical resources do not store `createTime`, as it is shared with the anchor.
 - `createTime` is needed when looking for something older than 30 days – that is, when there is no `beginTime` this old because the historical resources have timed out.
- **endTime** - The `endTime` timestamp records when the element was deleted.
 - All resources and historic resources that are representations of the same thing have a distinct `endTime`.
 - Resource `endTime` is used in historic graph traversals, that is, when the `_at` parameter is supplied.
 - For current resources, `endTime` is `LONG_MAX`. This is sometimes hidden through the REST API.
- **observedTime** - The `observedTime` timestamp records when the element was last observed, that is, when data was last input to the topology service for the element.

Topology examples

The following examples describe best-practice guidelines to provide you with practical information, such as implementation examples and code samples, that you can use to get started.

Restriction: The examples provided here must be amended and adjusted to suit your own implementation.

[Virtual machine example](#)

One of the most important goals of IBM Cloud Pak® for AIOps is to support the assurance and provisioning of modern IT, network, and storage environments. These environments all make extensive use of increasingly nested virtualization technologies that need to be modeled. The following example introduces such an IT Virtualization scenario, and describes an OpenStack response that provides a solution.

[Physical device example](#)

The following example of an IBM Tivoli Network Manager environment accessed through IBM Cloud Pak for AIOps provides insights into any environment that uses physical network devices.

Virtual machine example

One of the most important goals of IBM Cloud Pak® for AIOps is to support the assurance and provisioning of modern IT, network, and storage environments. These environments all make extensive use of increasingly nested virtualization technologies that need to be modeled. The following recipe introduces such an IT Virtualization scenario, and describes an OpenStack response that provides a solution.

IT Virtualization

The IBM Cloud Pak for AIOps model of a nested virtualization scenario can extend the traditional types of models provided by other solutions.

This model can represent a multi-domain view of the world that links IT, network, storage, applications, and services. In addition, it can incorporate concepts such as OpenStack's Heat Orchestration and Slack collaboration relative to traditional IT resources.

Some of the benefits of this approach are:

- To provide extra context

Increasingly, teams are more multi-disciplined and no longer operate in informational or functional silos. For example, network teams may include IT Virtualization specialists.

Such teams need access to more context when needed, to be able to answer some of their business-critical questions, such as:

- What storage volume is a VM attached to?
- Which orchestration step realized a network port?
- Who collaborated with whom for a particular incident?
- Which applications and services are supported by a network subnet?
- Which VM instances were shut down as part of a scale-in activity 1 hour ago?
- What is the impact of removing a Hypervisor from the environment?
- Which fixed IP addresses have a floating IP address that has been bound to in the last week?
- To provide a data-rich base

Added value services can be bolted onto a base system, provided the information exists, and the system has an architecture that allows for rapid extension.

For example, when building analytics on the topology data, the availability of information such as seasonality can provide extra insights.

The following diagram depicts the nested layers of virtualization, including networking, between these layers and technologies such as Docker and LXC or LXD.

Note: The services that are exposed can be applications or appear to be traditionally physical services such as network routers, switches, firewalls, and load balancers (a key goal of NFV).

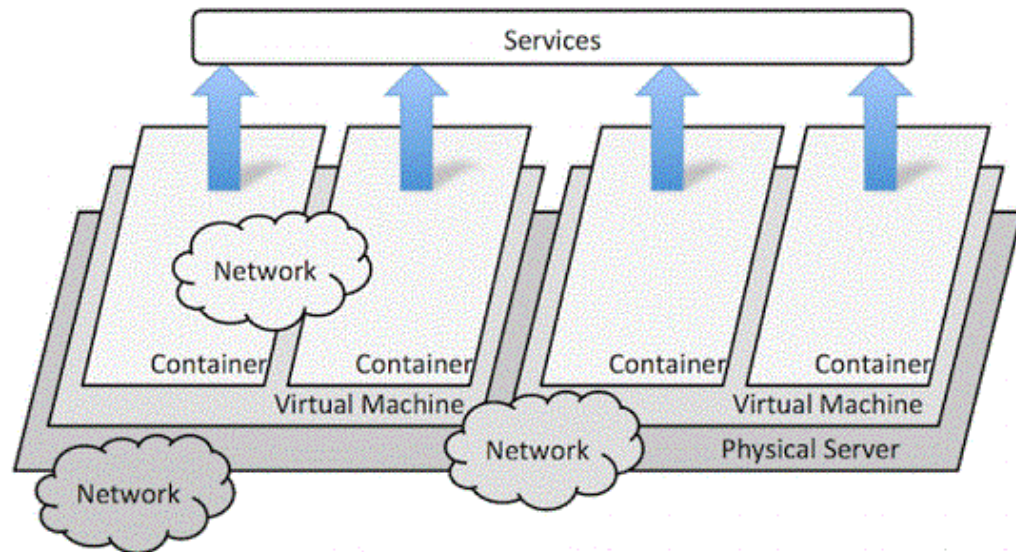


Figure. Physical services

OpenStack

OpenStack is a free and open source platform for cloud computing, typically deployed as an IaaS (Infrastructure-as-a-Service) capability. The software platform consists of interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and network resources throughout and between data centers.

OpenStack provides a number of projects, and related services and APIs, that are summarized here, as they speak directly to the need to have a multi-domain view of the environment. For more information, see the [OpenStack project navigator](#)

OpenStack **core services** include the following services:

- Nova

Compute manages the lifecycle of compute instances in an OpenStack environment.

- Neutron

Networking enables network connectivity as a service for other OpenStack services.

- Swift

Object Storage stores and retrieves arbitrarily unstructured data through a REST API.

- Cinder

Block Storage provides persistent storage to running instances.

- Keystone

Identity provides authentication and authorization services to OpenStack services and a service catalog.

- Glance

Image Service stores and retrieves virtual machine disk images for use by Nova.

OpenStack **optional services** include the following services:

- Horizon
dashboarding
- Ceilometer
telemetry
- Heat
orchestration
- Sahara
Platform Symphony MapReduce
- Designate
DNS
- Barbican
Key Management

IBM Cloud Pak for AIOps provides an Observer that makes extensive use of OpenStack's core and Heat APIs to build an end-to-end (or multi-domain) model.

[IT Virtualization OpenStack scenario](#)

The following example of an OpenStack environment accessed through IBM Cloud Pak for AIOps provides insights into any environment consisting of, for example, a combination of physical services and storage combined with virtualization from VMware (or similar).

IT Virtualization OpenStack scenario

The following example of an OpenStack environment provides insights into any environment consisting of, for example, a combination of physical services and storage combined with virtualization from VMware, or similar virtualization providing infrastructure.

IT virtualization

The following figure depicts a Hypervisor running a VM (LB Server) that is assigned to a tenant, has classification data and a number of network interfaces. Each network interface has a MAC address, one or more IP addresses (v4 or v6) and each IP address is considered to be part of an IP subnet. It is also related to orchestration data from Heat that helps identify how the instance was created.

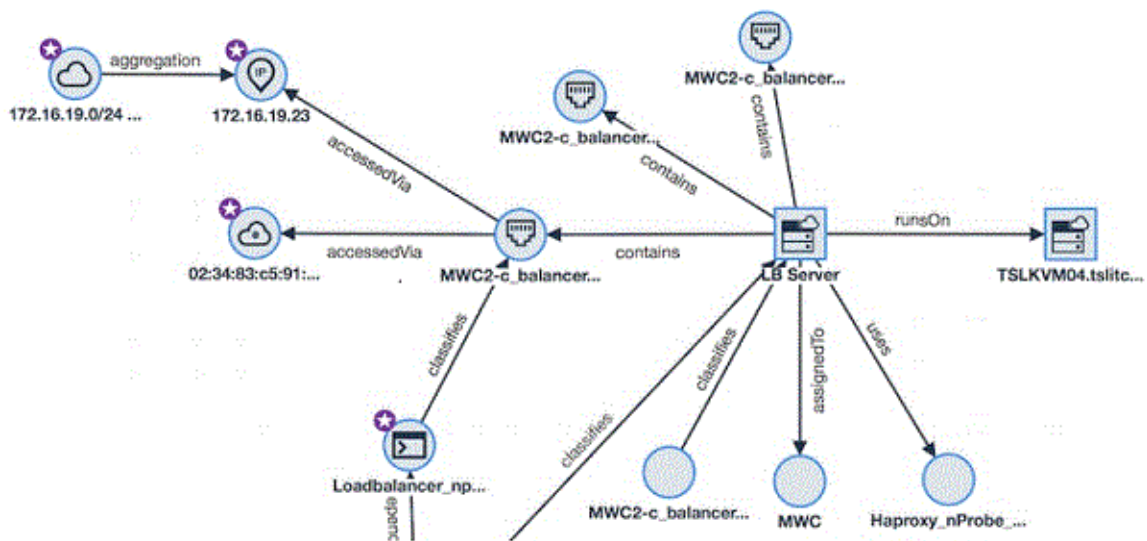


Figure. Example multi-domain topology

Property use guidance

Although the topology service is flexible, you should use the following guidelines when setting property values to ensure that elements are appropriately represented:

- Set **entityType** to one of the pre-defined entity types with an associated icon where possible.
- Set **name** and **description** to values that are easy to read or remember wherever possible.
- Use the default generic properties to represent generally applicable characteristics of an element and to provide a basic degree of labeling and filtering. For a list of generic properties, see the following topic: [Properties](#)

Model patterns - Part one

Stepping through each of the sections of the example of a multi-domain topology helps to identify reusable patterns.

In the following figure, the Hypervisor 'TSLKVM04' is running a VM 'LB Server'.

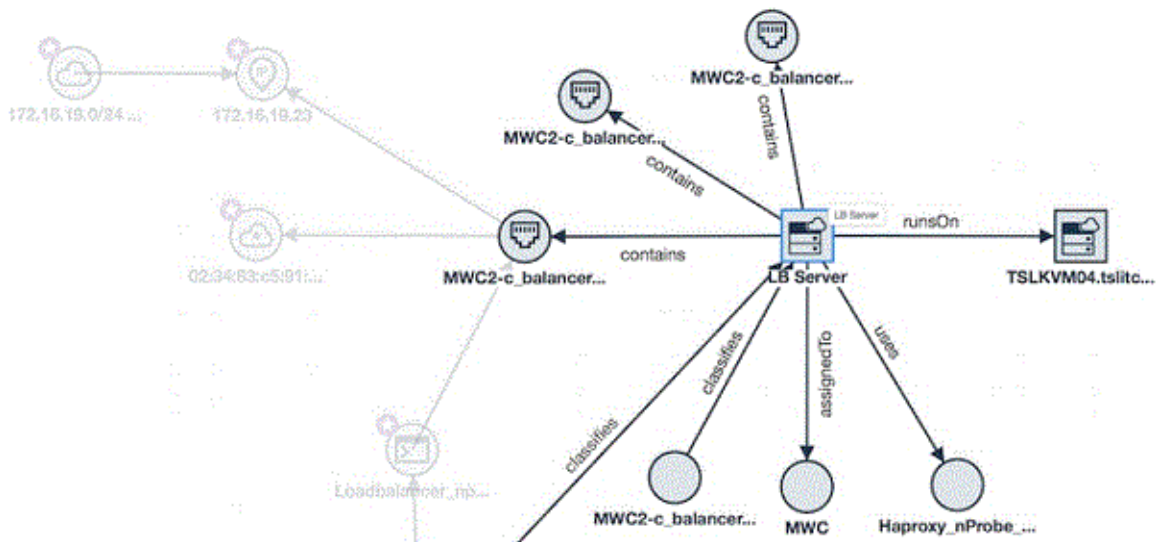


Figure. Figure of Hypervisor

The LB Server in this image is:

- Assigned to the 'MWC' tenant
- Contains three network interfaces
- Uses the Haproxy_nProbe image (OpenStack image)
- Is classified as an MWC2-c_balancer (OpenStack flavor)

This means that the following pattern can be identified:

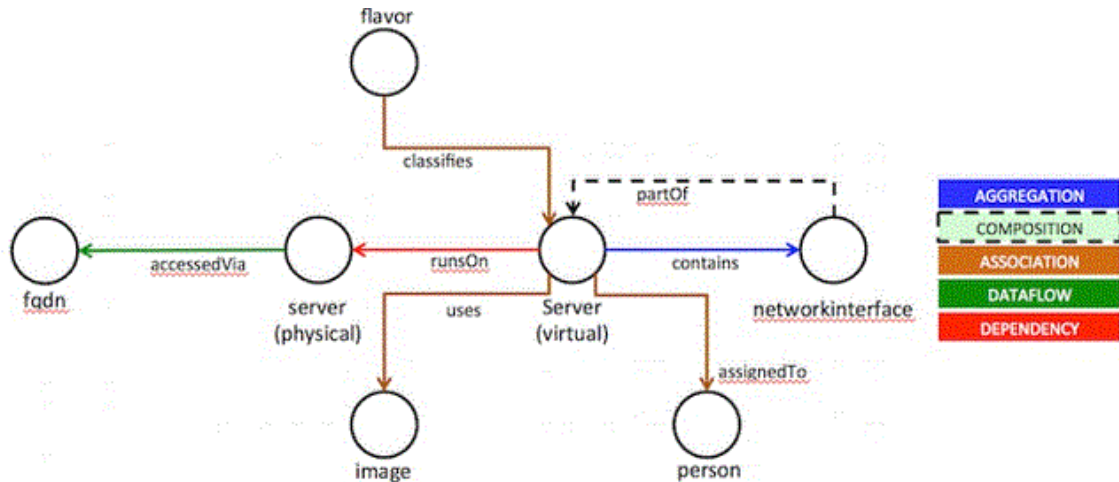


Figure. Pattern

Usage tips:

- The associations shown for flavor, image and person are optional.
 - Network interfaces and other components of the device must be associated with it through a **partOf** relationship. The exception is if an IP or MAC address is known independently of any device, for example flow data would expose those but the device would be unknown.
 - The FQDN (hostname, short or full DNS name) is associated directly with the server in this case as nothing else is known about the Hypervisor.
- Note:** This is **not** shown in the topology fragment.
- The **partOf** composition is not shown by the topology GUI, but must be created where the relationship between a component and a device is known. This is in addition to relationships such as **contains**, which the IBM Cloud Pak for AIOps console will show.

Hypervisor example JSON

```
{
  "_executionTime": 3,
  "createTime": 1501741671066,
  "name": "TSLKVM04",
  "uniqueId": "CYooventry_DC1:MWC/ComputeHost/TSLKVM04",
  "observedTime": 1501776262368,
  "_startedAt": "2017-08-03T06:27:51.066Z",
  "entityTypes": [
    "server"
  ],
  "beginTime": 1501741671066,
  "_id": "KNN6TCGhKyM4MCI6jooGWg",
  "_observedAt": "2017-08-03T16:04:22.368Z",
  "_modifiedAt": "2017-08-03T06:27:51.066Z",
  "_createdAt": "2017-08-03T06:27:51.066Z",
}
```

```

    "changeTime": 1501741671066,
    "matchTokens": [
      "Coventry_DC1:MWC/ComputeHost/TSLKVM04",
      "TSLKVM04"
    ]
  }
}

```

Note: Many of the properties starting with `_` are internal (such as timestamps). Also:

- The `uniqueId` is a composite of a number of fields: data center, tenant, className and name of the Hypervisor because the ID is a highly ambiguous integer.
- The `name` is the name of the instance from OpenStack.
- The `entityType` is set to 'server' to ensure correct classification and icon use.

Virtual machine example JSON

```

{
  "instanceName": "LB server",
  "tenantId": "2f79c691570c4a598be386325ea01da8",
  "launchedAt": 1501741751194,
  "_executionTime": 4,
  "userId": "48c7cd25ad0842be8e9b84390de0e587",
  "imageName": "None Available",
  "availabilityZone": "nova",
  "createTime": 1501741733817,
  "flavorName": "m1.nano",
  "name": "LB server",
  "uniqueId": "1e35c68a-86b0-445f-9741-e581121a0577",
  "serverStatus": "active",
  "observedTime": 1501741752717,
  "_startedAt": "2017-08-03T06:29:12.717Z",
  "entityTypes": [
    "server",
    "vm"
  ],
  "beginTime": 1501741752717,
  "flavorId": "42",
  "vmState": "active",
  "_id": "3nDmTkKfrvNhZinSDCYHDw",
  "_observedAt": "2017-08-03T06:29:12.717Z",
  "createdAt": 1501741733000,
  "_modifiedAt": "2017-08-03T06:29:12.717Z",
  "_createdAt": "2017-08-03T06:28:53.817Z",
  "changeTime": 1501741752717,
  "matchTokens": [
    "1e35c68a-86b0-445f-9741-e581121a0577",
    "LB server"
  ]
}

```

Note: Many of the properties starting with `_` are internal (such as timestamps). Also:

- The `uniqueId` in this case is the UUID of the instance from OpenStack.
- The `name` is the name of the instance from OpenStack.
- The `entityType` is set to 'server' and 'vm' to ensure correct classification and icon use.
- The `isVirtual` Boolean is set to true.

Network interface example JSON

```
{
  "_executionTime": 3,
  "isAdminStateUp": true,
  "createTime": 1501741717674,
  "name": "aab47c85-3110-401a-8bad-960b7c4bcd7b",
  "uniqueId": "aab47c85-3110-401a-8bad-960b7c4bcd7b",
  "observedTime": 1501741718855,
  "_startedAt": "2017-08-03T06:28:38.855Z",
  "entityTypes": [
    "networkinterface"
  ],
  "beginTime": 1501741718855,
  "isPortSecurityEnabled": true,
  "_id": "kkKpDH6yLn7cmNVXOddImg",
  "_observedAt": "2017-08-03T06:28:38.855Z",
  "_modifiedAt": "2017-08-03T06:28:38.855Z",
  "_createdAt": "2017-08-03T06:28:37.674Z",
  "changeTime": 1501741718855,
  "matchTokens": [
    "aab47c85-3110-401a-8bad-960b7c4bcd7b"
  ]
}
```

Note: As with the LB Server data, many of the properties are internal. Also:

- The **uniqueId** in this case is the UUID of the instance from OpenStack.
- The **name** is the name of the instance from OpenStack.
- The **entityType** is set to **networkinterface** to ensure correct classification and icon use.
- The type of the interface (such as **ifType** from IBM Tivoli Network Manager) is unknown.

Model patterns - Part two

Note: Each section of the example of the multi-domain topology identifies the reusable patterns.

In the following figure, the LB Server VM contains a network interface.

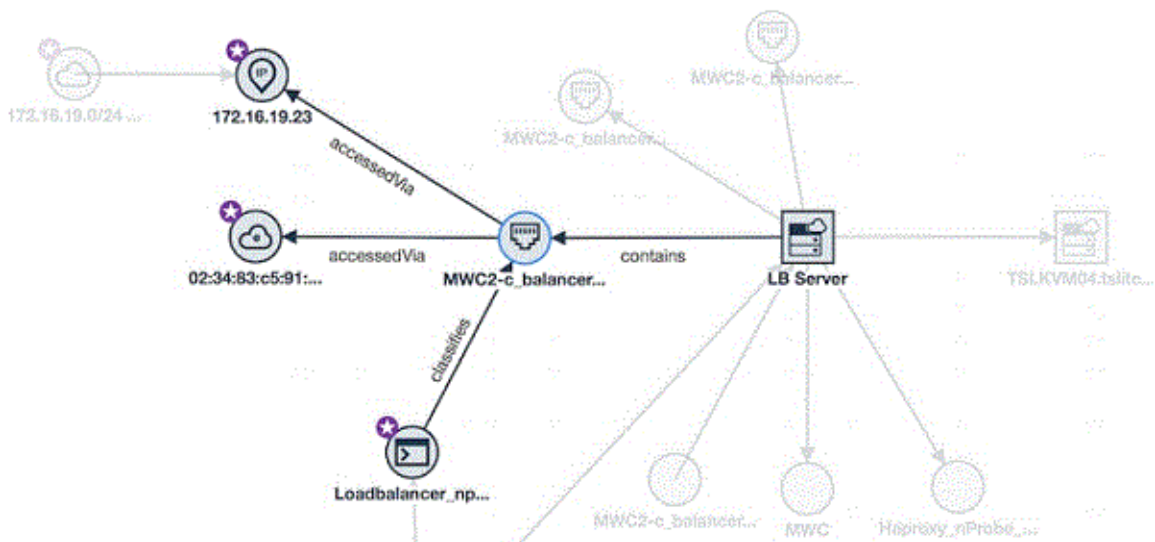


Figure. LB Server VM

The network interface in this image is:

- Accessed through an IP address and a MAC address

- Classified by a Heat orchestration resource

This means that the following pattern can be identified:

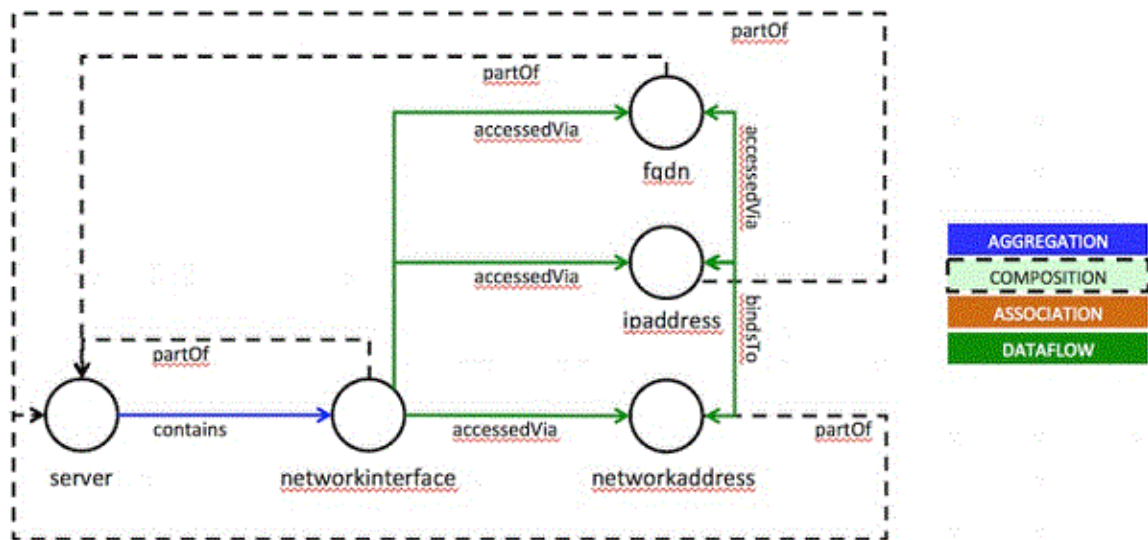


Figure. Pattern

Usage Tips:

- The Heat orchestration element and relationship is optional. Such things can be added if known to provide extra context.
- The network interface must be contained by and **partOf** the device. **Contains** is more fine-grained and may reference intermediate cards (for example) within the containment hierarchy of the device, such as

```
json device--contains-->card--->contains--->port
```
- If an IP address, FQDN or MAC address is known to be related to a specific device, then they must be associated with the device through a **partOf** relationship *in addition* to the **accessedVia** relationship.
- If an IP address, FQDN, or MAC address is known independently of a device, then no **partOf** relationship from them is necessary.
- If an IP address is known to resolve to an FQDN, then relationships between them should be created to depict that one resolves to another and one accesses another (**accessedVia** shown in the example).

IP address example JSON

```
{
  "_executionTime": 3,
  "createTime": 1501741717656,
  "name": "172.24.4.5",
  "uniqueId": "172.24.4.5",
  "ipNumber": 2887255045,
  "addressSpace": "Coventry_DC1:MWC",
  "observedTime": 1501741731565,
  "_startedAt": "2017-08-03T06:28:51.565Z",
  "entityTypes": [
    "ipaddress"
  ],
  "beginTime": 1501741731565,
  "version": "IPv4",
}
```



```

    "_id": "jPLc72DU-UvPeTQE_7YdPQ",
    "_observedAt": "2017-08-03T06:28:51.565Z",
    "_modifiedAt": "2017-08-03T06:28:51.565Z",
    "_createdAt": "2017-08-03T06:28:37.656Z",
    "changeTime": 1501741731565,
    "matchTokens": [
      "172.24.4.5",
      "IPv4:2887255045"
    ]
  }
}

```

Note:

- The **uniqueId** is the IP address itself. This is an RFC1918 IP address and so it **must** be qualified with an address space to disambiguate it from other instances of the same IP address. Similar precautions should be used with MAC addresses, which can be ambiguous.
- The **protocol** reflects whether the IP address is an IPv4 or IPv6 address.
- The **ipNumber** is a numeric representation of the IPv4 or IPv6 address. A Java BigInteger has the precision to represent IPv6 addresses.

Model patterns - Part three

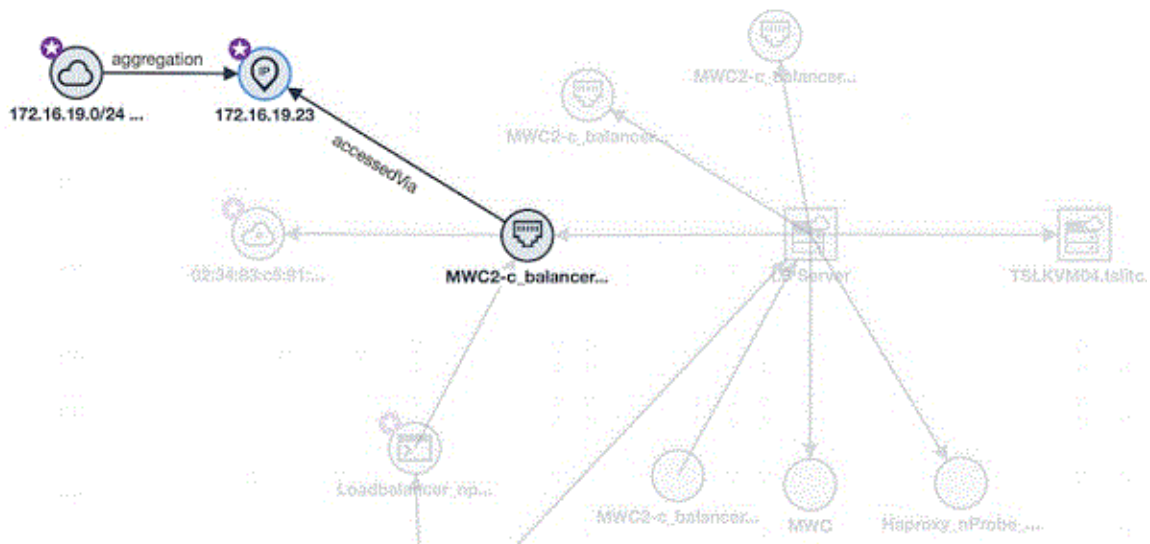


Figure. Multi-domain topology

This means that the following pattern can be identified:

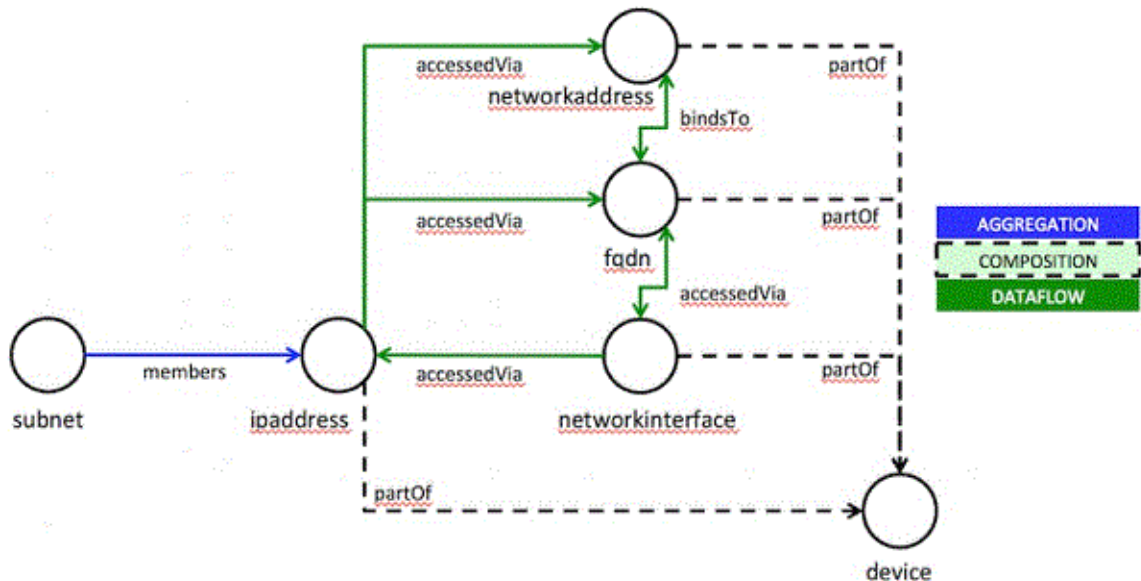


Figure. Pattern

Usage Tips:

- The IP subnet should aggregate any IP addresses known to be in it.
- Elements accessed through an IP should be related to it, such as a network interface, service, or process.
- If an FQDN is known to resolve to an IP address (and vice-versa), then they should be related.
- If a MAC address is known to bind to an IP address and vice-versa, they should be related.
- If an IP address, MAC address, or FQDN is known to relate to a device, they should be considered **partOf** it; otherwise, they are independent.

IP subnet example JSON

```
{
  "uniqueId": "c009ff59-13b9-48dc-8863-cd0c75070d99",
  "name": "172.24.4.0/24 (public-subnet)",
  "entityTypes": [
    "subnet"
  ],
  "matchTokens": [
    "172.24.4.0/24 (public-subnet)",
    "172.24.4.0/24",
    "c009ff59-13b9-48dc-8863-cd0c75070d99"
  ],
  "_id": "u0sjaHcumtg5A4DRl1fyAQ",
  "_references": [
    {
      "_id": "9duzx1-3o52g-ys5-47si0",
      "_edgeType": "contains",
      "_label": "aggregation",
      "_fromId": "u0sjaHcumtg5A4DRl1fyAQ",
      "_toId": "1sJQs8JmYzDQ-_wUOfvJbg",
      "_fromUniqueId": "c009ff59-13b9-48dc-8863-cd0c75070d99",
      "_toUniqueId": "172.24.4.12",
      "createTime": 1501838680418,
      "observedAt": "2017-08-04T09:24:40.418Z",
      "createdAt": "2017-08-04T09:24:40.418Z",
    }
  ]
}
```

```

    "beginTime": 1501838680418,
    "_startedAt": "2017-08-04T09:24:40.418Z",
    "observedTime": 1501838680418
  },
],
"_executionTime": 11,
"_modifiedAt": "2017-08-04T09:24:40.356Z",
"isDhcpEnabled": false,
"dnsNames": "None Available",
"_observedAt": "2017-08-04T09:24:40.356Z",
"gatewayIp": "172.24.4.1",
"_startedAt": "2017-08-04T09:24:40.356Z",
"observedTime": 1501838680356,
"changeTime": 1501838680356,
"ipVersion": "V4",
"createTime": 1501838680356,
"_createdAt": "2017-08-04T09:24:40.356Z",
"cidr": "172.24.4.0/24",
"networkId": "3e2b5d07-653a-4fc8-8224-45801d9d113f",
"beginTime": 1501838680356,
"allocationPools": "172.24.4.2-to-172.24.4.254"
}

```

Notes:

- The **uniqueId** in this case is the UUID of the subnet from OpenStack.
- The **name** in this case is set to CIDR notation plus the ID of the subnet.
- The **entityType** is set to subnet to ensure appropriate classification and icon usage.
- Some example relationships are shown: For example, an IP address that is part of the subnet is visible, and the subnet's use of an allocation pool is also depicted.

Physical device recipe

The following example of an IBM Tivoli Network Manager environment accessed through IBM Cloud Pak® for AIOps provides insights into any environment that uses physical network devices.

Network physical devices

The following figure depicts two physical devices, 172.20.1.104 and 172.20.1.1, which are connected to each other.

Server 172.20.1.1 has 3 Gigabit Ethernet ports on the same card, one of which has a subinterface with an **ifName** of 'ge-1/1/3.0'. That subinterface shares the MAC address as its physical parent and has two IPv6 addresses and one IPv4 address associated with it.

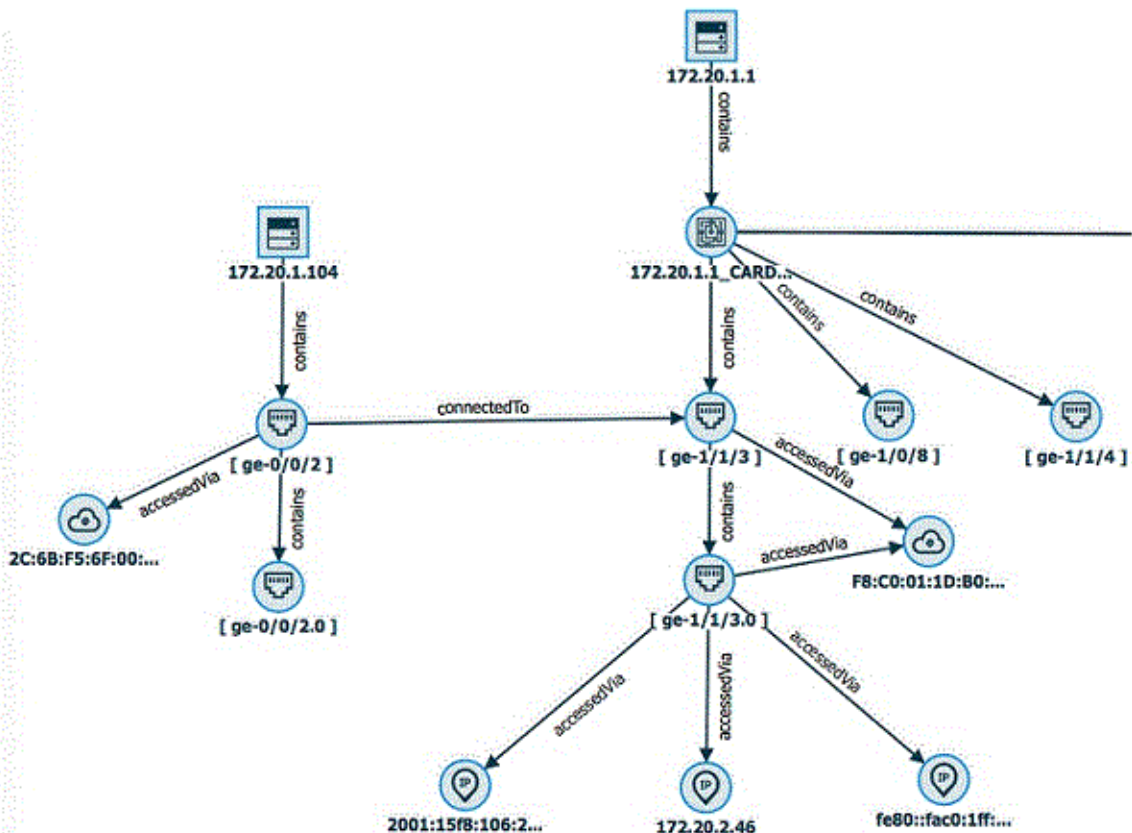


Figure. Figure that shows devices

Outline pattern

The topology service has an edge type of **partOf**, which is a member of the composition edge label family. For more information, see [Edge labels](#).

The following image illustrates how this relationship is used to tie the card, interfaces, and network addresses to the hosting device.

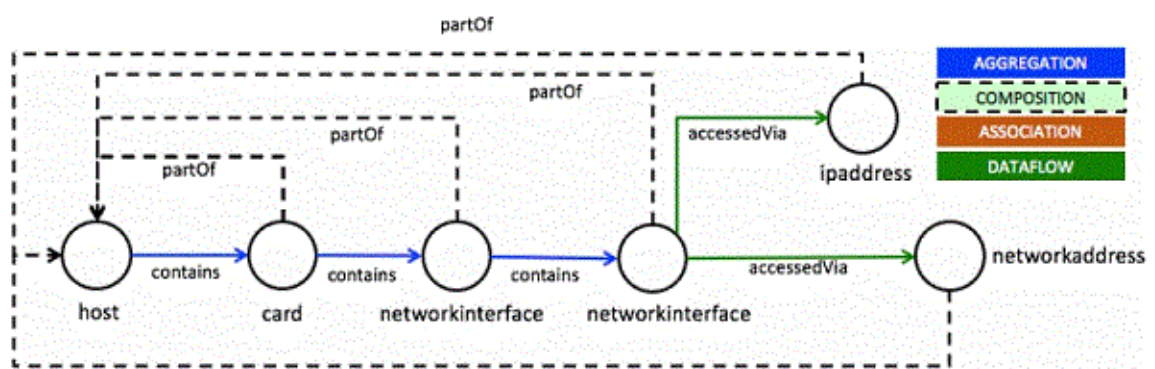
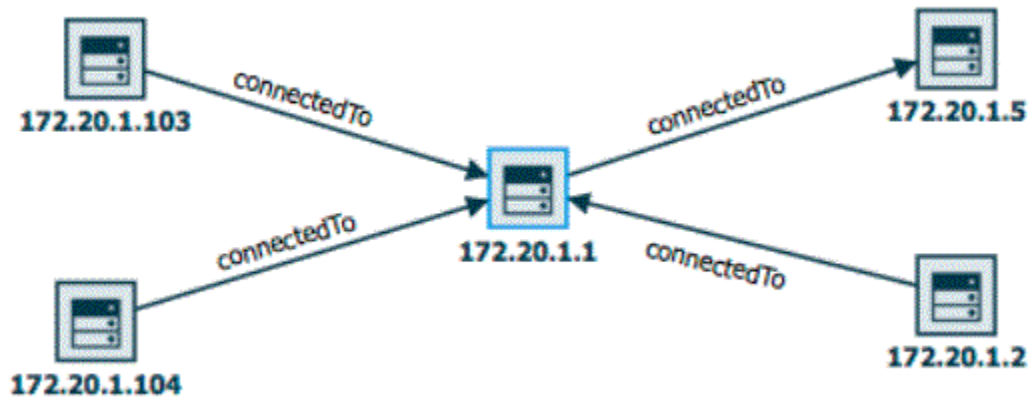


Figure. Relationships to hosting device

The **partOf** relationship is not shown as an explicit edge in the topology UI. However, it is used to determine which resources should be hidden in the host-to-host view, which in the context of this scenario would show just the hosts 172.20.1.104 and 172.20.1.1. It would also show the **connectedTo** edge between them, as depicted in the following part of the image.



{: caption="Figure.

Figure that shows **partOf** relationship" caption-side="bottom"}

Host example JSON

The following JSON extract is an example of the properties that you might choose to include when creating a host vertex. The properties that start with an underscore character are aliases for some of the read-only generic properties. They also include a few read/write generic properties, such as **name** and **uniqueId**. However, most of the properties in this example are free from 'User' properties.

```

{
  "uniqueId": "NCOMS:172.20.1.1",
  "name": "172.20.1.1",
  "entityTypes": [
    "host"
  ],
  "_createdAt": "2017-03-03T16:02:40.845Z",
  "_observedAt": "2017-03-03T16:04:18.636Z",
  "_id": "y7EXOKrHud21CWySCyMsBg",
  "_href": "/1.0/topology/resources/y7EXOKrHud21CWySCyMsBg",
  "_nodeType": "resource",
  "_executionTime": 4,
  "_modifiedAt": "2017-03-03T16:02:40.845Z",
  "matchTokens": [
    "sbk-pe1-jrmx80.southbank.eu.test.lab"
  ],
  "sysObjectId": "1.3.6.1.4.1.2636.1.1.1.2.90",
  "entityChangeTime": "2017-03-03T14:13:17.000Z",
  "className": "JuniperMSeries",
  "services": "datalink(2) network(3)",
  "_startedAt": "2017-03-03T16:02:40.845Z",
  "manual": 0,
  "cdmAdminState": 0,
  "cdmType": 2,
  "sysDescription": "Juniper Networks, Inc. mx5-t internet router, kernel JUNOS 15.1F4.15, Build date: 2015-12-23 20:50:37 UTC Copyright (c) 1996-2015 Juniper Networks, Inc.",
  "sysName": "sbk-pe1-jrmx80.southbank.eu.test.lab",
  "sysLocation": "The Mad Hatter Hotel 3-7 Stamford St London SE1 9NY UK,-0.10499474,51.50711477",
  "interfaceCount": 73,
  "entityCreateTime": "2017-03-03T14:13:17.000Z",
  "isIpForwarding": "forwarding",
  "accessIPAddress": "172.20.1.1",
  "entityDiscoveryTime": "2017-03-03T14:11:09.000Z",
  "sysContact": "williamking@uk.ibm.com",

```

```

    "_tenantIds": [
      "MoaldcmKHfx3dlyJnGm6JQ"
    ],
    "accessProtocol": "IPv4"
  }
}

```

Note: Some of the generic properties in this example are:

- The **uniqueId** in this case is a string, which uniquely identifies this resource to IBM Tivoli Network Manager.
- The **name** is the name of the host and will be used in the UI.
- The **entityType** is set to 'host'.
- The **matchTokens** contains the FQDN name of the host.

Network interface example JSON

```

{
  "uniqueId": "NCOMS:172.20.1.1[ ge-1/1/3.0 ]",
  "name": "[ ge-1/1/3.0 ]",
  "entityTypes": [
    "networkinterface"
  ],
  "_createdAt": "2017-03-03T16:03:03.889Z",
  "_observedAt": "2017-03-03T16:03:03.939Z",
  "_id": "v8aFVG6JoigYxTrlFSDkLw",
  "_href": "/1.0/topology/resources/v8aFVG6JoigYxTrlFSDkLw",
  "_nodeType": "resource",
  "_executionTime": 5,
  "operationalStatus": "started",
  "ifIndex": 537,
  "ifAdminStatus": "up",
  "_modifiedAt": "2017-03-03T16:03:03.940Z",
  "ifType": 53,
  "matchTokens": [
    "ge-1/1/3.0",
    "ifEntry.537"
  ],
  "ifName": "ge-1/1/3.0",
  "ifTypeString": "propVirtual",
  "connectorPresent": "false",
  "_startedAt": "2017-03-03T16:03:03.939Z",
  "speed": 1000000000,
  "mtu": 1500,
  "accessIpAddress": "172.20.2.46",
  "operationalDuplex": "FullDuplex",
  "promiscuous": false,
  "physicalAddress": "F8:C0:01:1D:B0:13",
  "ifDescription": "ge-1/1/3.0",
  "ifOperStatus": "up",
  "_tenantIds": [
    "MoaldcmKHfx3dlyJnGm6JQ"
  ],
  "accessProtocol": "IPv4"
}

```

Note: Some of the generic properties in this example are the following:

- The **uniqueId** in this case is a string, which uniquely identifies this resource to IBM Tivoli Network Manager.
- The **name** is the name of the host and will be used in the UI.
- The **entityType** is set to **networkinterface**.

- The `matchTokens` contains both the `ifName` and a string denoting the `ifIndex` in the `ifEntry` table.

Infrastructure Management API Documentation

In this guide you will find a detailed overview of REST API, reference material, appendices, and examples. The guides provide information on how to integrate Infrastructure Automation – Infrastructure Management with external applications. It details the specification of the Infrastructure Management REST API, which is implemented as standard REST HTTP requests and responses of content type JSON.

- [API Overview](#)
- [API Reference](#)
- [API Appendices](#)
- [API Examples](#)
- [Prerequisites](#)

Infrastructure Management API Overview

In this guide you will find a detailed overview of Infrastructure Automation – Infrastructure Management REST API.

- [REST API Authorization](#)
- [Password Updates](#)
- [CRUD](#)
- [HTTP Basics](#)
- [JSON Specification](#)
- [Query Specification](#)
- [Bulk Querying](#)
- [Collection Filtering](#)
- [Pagination](#)

Prerequisites for Infrastructure Management APIs

Infrastructure Management can be installed as either a VM appliance or as a containerized deployment. To issue API calls to the Infrastructure Management that is installed as a containerized deployment, obtain the authorization token and the URL hostname as described below.

Set the values for the following variables:

Variable	Description
IM_USER	IM_USER is the user you want to use. You can use either the default admin user or any user in the user group that you have configured in the im-iminstall CR. See https://ibm.biz/IM_containerized_pod .
IM_PASSWORD	Set the user password variable.

Variable	Description
CP_CONSOLE_HOSTNAME	Set the CP console hostname with the command <code>oc -n <IA_installation_namespace> get routes cp-console -o=jsonpath='{.spec.host}'</code> .

Where `<IA_installation_namespace>` is the namespace that Infrastructure Automation is installed in.

1. Run the command to get the bearer token for the user

```
export IM_BEARER_TOKEN=$(curl -ks -X POST -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" -d "grant_type=password&username=${IM_USER}&password=${IM_PASSWORD}&scope=openid" https://${CP_CONSOLE_HOSTNAME}/idprovider/v1/auth/identitytoken|jq -r .access_token)
```

2. Run the command to get the URL hostname,

```
export IM_HOSTNAME=$(oc -n cp4aiops get route | grep inframgmt | awk '{print $2}')
```

3. After the variables `IM_BEARER_TOKEN` and `IM_HOSTNAME` are defined, use them to issue the API calls. A curl command should be in this format,

```
curl -H "Authorization: Bearer ${IM_BEARER_TOKEN}" \
..... \
"https://${IM_HOSTNAME}/api/....."
```

For example, to issue a GET request to the Infrastructure Management that is installed as a containerized deployment, specify,

```
curl -H "Authorization: Bearer ${IM_BEARER_TOKEN}" \
-k -i -X GET -H "Accept: application/json" \
"https://${IM_HOSTNAME}/api/providers"
```

Note: To issue API calls to the Infrastructure Management that is installed as a VM appliance, follow the instructions in the subsequent sections in Infrastructure Management API Guide. To issue API calls to the Infrastructure Management that is installed as a containerized deployment, replace the HTTP authorization header and the URL segment (<http://localhost:3000>) with those described above.

- [REST API Authorization](#)
 - [Specifying Authorization Group](#)
 - [Specifying an Invalid Authorization Group](#)

REST API Authorization

When authenticating to the appliance, the current group of the authenticated user will be used for determining the authorization of what the user can see and do using the REST API. While this is acceptable when users only belong to a single group, this falls short when users are authenticating using their External Authentication credentials (IPA, LDAP or AD) where they may belong to multiple groups.

Specifying Authorization Group

This section describes how to specify an authorization group for the user. The mechanism described here is honored for both methods of authentication:

- Basic Authentication
- Token based Authentication

The user's group is to be specified in the new HTTP Header:

X-MIQ-Group

and is to contain the user's group name.

Example request using Basic Authentication:

```
$ curl --user username:password
  -i -X GET -H "Accept: application/json"
  -H "X-MIQ-Group: auditors"
  http://localhost:3000/api/vms
```

Example request using Token based Authentication:

```
$ curl -i -X GET -H "Accept: application/json"
  -H "X-Auth-Token: 93d98fd44cc5a401ff385ba87b21fd13"
  -H "X-MIQ-Group: auditors"
  http://localhost:3000/api/vms
```

The validated users's group will be in effect for the duration of the REST API request. Multiple requests can be made as a particular user but with different authorization groups concurrently.

Specifying an Invalid Authorization Group

In case an invalid authorization group is specified, a 401 Unauthorized is returned:

```
HTTP/1.1 401 Unauthorized
X-Frame-Options: SAMEORIGIN
...
{
  "error" : {
    "kind" : "unauthorized",
    "message" : "Invalid Authorization Group <invalid_group_name> specified",
    "klass" : "ApiController::AuthenticationError"
  }
}
```

- [Password Updates](#)

Password Updates

When the appliance is configured for Database authentication, users can update their passwords via the REST API.

Note that if an authenticated REST API user is entitled to edit users, they can edit other users' passwords, e.g. admin account.

Passwords can be updated via the **edit** post action on the user's resource as follows:

```
POST /api/users/:id
{
  "action" : "edit",
  "resource" : { "password" : "<new_password>" }
}
```

So for updating one's password, the `/api/users/:id` being updated must be the authenticated REST API user's object.

- [CRUD Request Examples](#)
 - [Show a collection of Resources](#)

- [Return a single Resource](#)
- [Create a Resource](#)
- [Update a Resource](#)
- [Delete a Resource](#)

CRUD Request Examples

The following examples show the basic CRUD operations (Create, Read, Update, Delete) using the REST API.

For Authentication, we are showing here basic authentication via the `-user admin:smartvm` credentials argument. For multiple API calls it is recommended to access the REST API via token based authentication, please refer the the [Authentication Page](#) for details.

Show a collection of Resources

Get a collection of services: **GET /api/services**

```
curl --user admin:smartvm
-i -X GET -H "Accept: application/json"
http://localhost:3000/api/services
```

Return a single Resource

Return a single service: **GET /api/services/:id**

```
curl --user admin:smartvm
-i -X GET -H "Accept: application/json"
http://localhost:3000/api/services/1
```

Create a Resource

Create a new provider: **POST /api/providers**

```
curl --user admin:smartvm
-i -X POST -H "Accept: application/json"
-d '{
  "type"      : "ManageIQ::Providers::Redhat::InfraManager",
  "name"      : "RHEVM Provider",
  "hostname"  : "rhevm.local.com",
  "ipaddress" : "192.168.5.1",
  "credentials" : {
    "userid"   : "admin",
    "password" : "12345"
  }
}'
http://localhost:3000/api/providers
```

Update a Resource

Update the name of a service: **PUT /api/services/:id**

```
curl --user admin:smartvm
-i -X PUT -H "Accept: application/json"
-d '{ "name" : "updated service name" }'
http://localhost:3000/api/services/1
```

Delete a Resource

Delete a service: **DELETE /api/services/:id**

```
curl --user admin:smartvm
  -i -X DELETE -H "Accept: application/json"
  http://localhost:3000/api/services/1
```

- [HTTP Basics](#)
- [REST API Entry Point](#)
 - [Response:](#)
 - [Entry Point Details](#)
- [Supported Content Types](#)
- [URL Paths](#)
- [Methods and related URLs](#)
- [Updating resources](#)
- [Modifying Resource attributes](#)
- [Return Codes](#)
 - [Success](#)
 - [Client Errors](#)
 - [Server Errors](#)

HTTP Basics

- [REST Entry Point](#)
 - [Entry Point Details](#)
- [Supported Content Types](#)
- [URL Paths](#)
- [Methods and related URLs](#)
- [Updating resources](#)
- [Return Codes](#)

REST API Entry Point

The REST API is made available via the /api URL prefix. It can be accessed on the Appliance server directly via HTTP using the default port as follows:

```
http://localhost:3000/api
```

Or external to the Appliance as follows:

```
https://<applianceHost_fqdn>/api
```

Response:

```
{
  "name" : "API",
  "description" : "REST API",
  "version" : "2.3.0",
  "versions" : [
    {
      "name" : "2.3.0",
      "href" : "http://localhost:3000/api/v2.3.0"
    }
  ],
}
```

```

...
"collections" : [
  ...
  {
    "name" : "providers",
    "href" : "http://localhost:3000/api/providers",
    "description" : "Providers"
  },
  ...
  {
    "name" : "vms",
    "href" : "http://localhost:3000/api/vms",
    "description" : "Virtual Machines"
  },
  ...
]
}

```

- Version is the Current API version, accessible via either of the following:
- /api/
- /api/v2.0.0/
- Versions lists current and any earlier API versions that are still exposed via their respective entry points:
- /api/vVersion/

Entry Point Details

An authenticated request of the entry point:

GET /api

returns the following structure:

```

{
  "name": "API",
  "description": "REST API",
  "version": "2.3.0",
  "versions": [
    {
      "name": "2.3.0",
      "href": "http://localhost:3000/api/v2.3.0"
    }
  ],
  "settings": {
  },
  "identity": {
    "userid": "admin",
    "name": "Administrator",
    "user_href": "http://localhost:3000/api/users/1",
    "group": "EvmGroup-super_administrator",
    "group_href": "http://localhost:3000/api/groups/2",
    "role": "EvmRole-super_administrator",
    "role_href": "http://localhost:3000/api/roles/1",
    "tenant": "My Company",
    "groups": [
      "EvmGroup-super_administrator"
    ]
  },
  "server_info": {

```

```

    "version": "master",
    "build": "20161104133757_01d7244",
    "appliance": "EVM"
  },
  "product_info": {
    "name": "ManageIQ",
    "name_full": "ManageIQ",
    "copyright": "Copyright (c) 2014-present ManageIQ Authors.",
    "support_website": "http://www.manageiq.org",
    "support_website_text": "ManageIQ.org"
  },
  "collections": [
    {
      "name": "actions",
      "href": "http://localhost:3000/api/actions",
      "description": "Actions"
    },
    ...
    {
      "name": "vms",
      "href": "http://localhost:3000/api/vms",
      "description": "Virtual Machines"
    },
    {
      "name": "zones",
      "href": "http://localhost:3000/api/zones",
      "description": "Zones"
    }
  ]
}

```

Where **settings** and **identity** are specific to the authenticated user.

Supported Content Types

Requests:

Accept: application/json

Responses:

Content-Type: application/json

URL Paths

The recommended convention for URLs is to use alternate collection / resource path segments, relative to the API entry point as described in the following example:

URL	Description
/api	The REST API entry point
/api/v Version	The REST entry point for a specific version of the REST API
/api/:collection	A top-level collection
/api/:collection/:id	A specific resource of that collection
/api/:collection/:id/:subcollection	Sub-collection under the specific resource

Methods and related URLs

The basic HTTP Methods used for the API are GET, POST, PUT, PATCH, OPTIONS and DELETE.

URL	Semantic
GET /api/:collection	Return all resources of the collection
GET /api/:collection/:id	Return the specific resource
POST /api/:collection	Create a resource in the collection
POST /api/:collection/:id	Perform an Action on a resource in the collection
PUT /api/:collection/:id	Update a specific resource
PATCH /api/:collection/:id	Update a specific resource
OPTIONS /api/:collection	Return metadata about the collection Please refer to Collection Metadata for details
DELETE /api/:collection/:id	Delete a specific resource

There :collection represent specific Appliance entities like services, hosts, vms, etc.

Updating resources

As shown in the above table, there are a couple of way to update attributes in a resource. These are:

- Update a resource via the PUT HTTP Method
- Update a resource via a POST Method with an *edit* action.
- Update a resource via the PATCH HTTP Method

While the PUT is the common method, the PATCH mechanism gives better control on which attribute to edit, add as well as remove which is not available with the other two methods.

Modifying Resource attributes

PUT /api/services/42

```
{
  "name" : "The new Service name",
  "description" : "A Description for the new Service"
}
```

POST /api/services/42

```
{
  "action" : "edit",
  "resource" : {
    "name" : "A new Service name",
    "description" : "A Description for the new Service"
  }
}
```

PATCH /api/services/42

```
[
  { "action": "edit", "path": "name", "value": "A new Service name" },
  { "action": "add", "path": "description", "value": "A Description for the new Service" },
  { "action": "remove", "path": "display" }
]
```

Return Codes

- [Success](#)
- [Client Errors](#)
- [Server Errors](#)

Success

- **200 OK** - The request has succeeded without errors, this code should be returned for example when retrieving a collection or a single resource.
- **201 Created** - The request has been fulfilled and resulted in a **new resource being created**. The resource is available before this status code is returned. The response includes the HTTP body of the newly created resource.
- **202 Accepted** - The request has been accepted for processing, but the processing has not been completed. Like, resource is not fully available yet. This status code is usually returned when the resource creation happens asynchronously. In this case the HTTP response includes a pointer to *monitor* or a *job* where the client can query to get the current status of the request and the estimate on when the request will be actually fulfilled.
- **204 No Content** - The server has fulfilled the request but does not need to return an entity-body, and might want to return updated meta information. This HTTP response is commonly used for the DELETE requests, as the resource that was deleted does not exist anymore.

Client Errors

- **400 Bad Request** - The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications. In REST API this status code should be returned to client when the client use the wrong combination of attributes, like expanding the non-existing collection, or using the pagination parameter incorrectly. Another use-case could be creating or performing actions on the resource, when the wrong JSON serialization of the resource or action is used.
- **401 Unauthorized** - The request requires user authentication. The response MUST include a *Authenticate* header field containing a challenge applicable to the requested resource. If the request include *Authenticate* header, then this HTTP status code might indicate that the current user is **not authorized** to perform given action or to access given resource.
- **403 Forbidden** - The server understood the request, but is refusing to fulfill it. Authorization will not help in this case. This HTTP status code might indicate that the action performed is not supported for this resource or collection.
- **404 Not Found** - In this case, the server has not found anything that matches with the URL.
- **415 Unsupported Media Type** - The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method. This error must be returned, when the client is explicitly asking for format other than JSON (application/json).

Server Errors

- **500 Internal Server Error** - The server encountered an unexpected condition which prevented it from fulfilling the request. This error code must be used when an exception is raised in the application and

the exception has nothing to do with the client request.

- [JSON Specification](#)
 - [General Conventions](#)
 - [Basic types](#)
 - [Common Attributes and Actions](#)
 - [Attributes](#)
 - [Actions](#)
 - [About permissions and security:](#)
 - [Collections](#)
 - [Action Specification](#)
 - [Collection actions](#)
 - [Resource actions](#)
 - [Action Responses](#)

JSON Specification

General Conventions

The API uses [JSON](#) throughout; the [Content-Type](#) for all requests and responses is **application/json**.

As is general practice with REST, clients should not make assumptions about the server's URL space. Clients are expected to discover all URL's by navigating the API. To keep this document readable, we still mention specific URL's, generally in the form of an absolute path. Clients should not use these, or assume that the actual URL structure follows these examples, and instead use discovered URL's. Any client should start its discovery with the API entry point, here denoted with **/api**.

- [Basic Types](#)
- [Common Attributes and Actions](#)
 - [Attributes](#)
- [Actions](#)
- [Collections](#)
- [Action Specification](#)
- [Collection Actions](#)
- [Resource Actions](#)
- [Action Responses](#)

Basic types

The following are basic data types and type combinators that are used throughout:

Name	Explanation	Example serialization
Integer	Integer value	{ "memsize" : 2048 }
String	JSON string	{ "state" : "running" }
URL	Absolute URL	{ "href" : "http://SERVER/vms/1/start" }
Timestamp	Timestamp in ISO8601 format	{ "created" : "2013-12-05T08:15:30Z" }
Array[T]	Array where each entry has type T	{ "vms" : [{ "id" : "1" }, { "id" : "2" }] }

Name	Explanation	Example serialization
Ref[T]	A reference to a T, used to model relations, the T is a valid Resource identifier	{ "vm" : { "href" : URL } }
Collection	Array[T] where T represents a Ref[T], this might allow actions to be executed on all members as a single unit	{ "vms" : { "count" : 2, "resources" : [{ "href" : URL }, { "href" : URL }], "actions" : [] } }
Struct	A structure with sub-attributes	"power_state": { "state" : "ON", "last_boot_time" : "2013-05-29T15:28Z", "state_change_time" : "2013-05-29T15:28Z" }

Common Attributes and Actions

The following describes attributes and actions that are shared by all resources and collections defined in this API.

Attributes

Attribute	Type	Description
id	String	A string identifier for the referenced resource
href	Ref(self)	A unique self reference
name	String	A human name of the resource

```
{
  "id" : "10",
  "href" : "http://localhost:3000/api/vms/10",
  "name" : "first_vm"
}
```

Actions

Action	HTTP method	Description
create	POST	Create new resource in the collection
edit	PUT/PATCH/POST	Edit attributes in resource
delete	DELETE	Delete resource

Note:

About permissions and security:

Advertising of the common actions depends purely on the role and permissions of that the current API user does have for the particular resource.

Collections

Resources can be grouped into collections. Each collection is homogeneous so that it contains only one type of resource, and is unordered. Resources can also exist outside any collection. In this case, we refer to these resources as singleton resources. Collections are themselves resources as well.

Collections can exist globally, at the top level of an API, but can also be contained inside a single resource. In the latter case, we refer to these collections as sub-collections. Sub-collections are usually used to express some kind of “contained in” relationship

Collections are serialized in JSON in the following way:

```
{
  "name" : String,
  "count": Integer,
  "subcount": Integer,
  "resources": [ ... ],
  "actions": [ ... ]
}
```

Where the **name** attribute is the collection name. The **count** attribute in a collection always denotes the total number of items in the collection, not the number of items returned. **subcount** attribute in a collection depicts the number of items returned. Then the **resources** attribute is an Array[T] where T might be a list of references to the T or, if expanded a list of resources with all attributes. The **actions** then contains an Array of actions that can be performed against the collection resources.

Action Specification

The representation of each resource will only contain an action and its URL if the current user is presently allowed to perform that action against that resource. Actions will be contained in the **actions** attribute of a resource; that attribute contains an array of action definition, where each action definition has a rel, method and a href attribute.

- **name** attribute contains the action name
- **method** attribute states the HTTP method that must be used in a client HTTP request in order to perform the given action (eg. GET, POST, PUT, DELETE)
- **href** attribute contains the absolute URL that the HTTP request should be performed against
- **form** an optional attribute that references a JSON document which describes the resource attributes that can be provided in the message body when performing this action. This description will indicate which of those attributes are mandatory and which are optional.

Collection actions

The actions performed against a collection of resources are in most cases batch operations against multiple resources. The action request must include an HTTP body with the action name and the list of resource representations that the action will be performed against.

The resource representation might include the resource attributes as they can change the way how the action is actually performed. In the example below, the first service is retired immediately, versus the second being retired at a later date with a retirement warning of 3 days.

Sample JSON request body for collection action:

POST /api/services

```
{
  "action": "request_retire",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/101" },
    { "href" : "http://localhost:3000/api/services/102",
      "date" : "10/30/2015",
      "warn" : 3
    }
  ]
}
```

Actions in collection:

```

{
  "name" : String,
  "count": Integer,
  "resources": [ ... ],
  "actions": [
    {
      "name" : "start",
      "method" : "post",
      "href" : URL
    },
    {
      "name" : "stop",
      "method" : "post",
      "href" : URL
    },
    {
      "name" : "suspend",
      "method" : "post",
      "href" : URL
    },
    {
      "name" : "edit",
      "method" : "post",
      "href" : URL
    },
    {
      "name" : "delete",
      "method" : "post",
      "href" : URL
    },
    {
      "name" : "delete",
      "method" : "delete",
      "href" : URL
    }
  ]
}

```

Resource actions

An action performed against a given resource is always described in the body of the HTTP request. The HTTP body could contain a list of resource attributes that dictate how the state of the receiving resource is to be changed once the action is performed. At minimum the JSON document in the message body must contain the name of the action to be performed.

In cases where no attributes are required to perform an action the HTTP body will contain an empty JSON document, in which case default values will be assigned to the corresponding attributes.

Sample JSON request body for resource action:

POST /api/services/101

```

{
  "action" : "request_retire",
  "resource" : { "date" : "10/30/2015", "warn" : 5 }
}

```

POST /api/vms/321

```

{
  "action" : "start",
  "resource" : {}
}

```

or Simply:

```
{
  "action" : "start"
}
```

Actions in a resource:

```
{
  "id" : String,
  "href" : Ref(self),
  "name" : "resource human name",
  "actions" : [
    {
      "name" : "edit",
      "method" : "post",
      "href" : URL
    }
  ]
}
```

Action Responses

When performing actions on resources, there are two types of responses that one is to expect.

1. For actions that operate on the resource itself like a *create* or *edit*, the response is usually the updated resource. This includes creation of Provision or Automate requests where the created `/api/provision_requests` and `/api/automation_requests` gets returned.
2. For others like a *start* or *stop* action, the response includes an action result for each targetted resource. An action result has the following construct in the response:

```
{
  "results" : [
    {
      "success" : true | false,
      "message" : String,
      "href" : Ref[resource],
      "result" : Struct,
      "task_id" : Id,
      "task_href" : Ref[task]
    },
    ...
  ]
}
```

results being an array of action results as one or more resources could be targeted in a request.

NOTE:

- success and message are always there.
- result is optional and would exist when an action results in data, i.e. policy resolve.
- href is populated for the resource being targeted by the action
- task_id and task_href are optional. They are defined when actions are run asynchronously and a task is created, i.e. <http://localhost:3000/api/tasks/:id> which can be monitored for action completion.

Other action specific attributes could also be returned in the the action result.

For tagging actions:

```
{
  "tag_category" : String,
  "tag_name" : String,
  "tag_href" : Ref[tags]
}
```

When executing actions on subcollections, like *policies*, *policy_profiles* and *service_templates*, the following is also provided in the action result:

```
{
  "<subcollection>_id" : Id,
  "<subcollection>_href" : Ref[subcollection]
}
```

- [Query Specification](#)
 - [Control Attributes](#)
 - [Filtering](#)
 - [Expanding Collection](#)
 - [Virtual Attributes and Relationships](#)

Query Specification

This specification identifies the controls available when querying collections.

The controls are specified in the GET URL as attribute value pairs as follows:

```
GET /api/:collection?attr1=val1&attr2=val2&...
```

Control Attributes

Category	Attribute	Semantics
Pagination		
	offset	0-based offset of first item to return
	limit	number of items to return. If 0 is specified then the remaining items are returned
	Please refer to Pagination for details on default pagination and pagination links provided by the API	
Scope		
	filter[]	One or more filters to search on. See Filtering below.
	attributes=atr1,atr2,...	Which attributes in addition to id and href to return. If not specified or <i>all</i> (default is <i>attributes=all</i>), then all attributes are returned
	expand=resources	To expand the resources returned in the collection and not just the href. See Expanding Collection below
Sorting		
	sort_by=atr1,atr2,...	By which attribute(s) to sort the result by. SQL friendly virtual attributes are also supported.
	sort_order=asc or desc	Order of the sort
	sort_options=ignore_case	Option for case insensitive sort

- The **count** attribute in a collection always denotes the total number of items in the collection, not the number of items returned.
- The **subcount** attribute in a collection denotes the number of items from the collection that were returned. For example, as a result of a paged request.

Filtering

GET requests against collections support the following query parameters to enable filtering:

- **filter**: The filter to use for querying the collection, i.e. filter[]=name='myservice%'.

GET /api/:collection?filter[]=name='myservice%'

For more information on filter[] please reference the [Filtering](#) page

Expanding Collection

While in the JSON serialization example the description says that the resource might be a list of references to these resource, using the **expand** parameter they can be expanded to return a full JSON serialization of the resource instead:

GET /api/vms

```
{
  "name" : "vms",
  "count" : 2,
  "subcount" : 2,
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/1" },
    { "href" : "http://localhost:3000/api/vms/2" }
  ],
  "actions" : []
}
```

GET /api/vms?expand=resources

```
{
  "name" : "vms",
  "count" : 2,
  "subcount" : 2,
  "resources" : [
    {
      "id" : "1",
      "href" : "http://localhost:3000/api/vms/1",
      "name" : "My First VM",
      ...
    },
    {
      "id" : "2",
      "href" : "http://localhost:3000/api/vms/2",
      "name" : "My Second VM",
      ...
    }
  ],
  "actions" : []
}
```

Virtual Attributes and Relationships

When querying resources, virtual attributes (not database columns) as well as relationships can be queried via the attributes parameter.

For example, while accounts and software are defined subcollections of vms

```
GET /api/vms/166?expand=software&attributes=ipaddresses,lans,storage
```

```
{
  "href": "http://localhost:3000/api/vms/166",
  "id": "166",
  "vendor": "vmware",
  "name": "Dev Nightly Appl",
  "location": "Nightly Appl 201.vmx",
  "host_id": "4",
  "created_on": "2014-11-20T19:37:28Z",
  "updated_on": "2015-03-07T04:32:33Z",
  "storage_id": "12",
  "guid": "a9409862-70ec-11e4-90c6-b8e85646e742",
  "uid_ems": "422f50c6-2ba6-1059-338d-423cc3daf5b4",
  "tools_status": "toolsNotRunning",
  "standby_action": "checkpoint",
  "power_state": "off",
  "state_changed_on": "2014-11-20T19:37:28Z",
  "connection_state": "connected",
  "memory_reserve": 0,
  "memory_reserve_expand": false,
  "memory_limit": -1,
  "memory_shares": 40960,
  "memory_shares_level": "normal",
  "raw_power_state": "poweredOff",
  ...
  "ipaddresses": [
    "192.168.100.1"
  ],
  "lans": [
    {
      "id": "8",
      "switch_id": "6",
      "name": "VM Network",
      "tag": "0",
      "created_on": "2014-11-20T19:37:23Z",
      "updated_on": "2014-11-20T19:37:23Z",
      "uid_ems": "VM Network",
      "computed_allow_promiscuous": false,
      "computed_forged_transmits": true,
      "computed_mac_changes": true
    }
  ],
  "storage": {
    "id": "12",
    "name": "StarM1-Dev",
    "store_type": "VMFS",
    "total_space": 2134061875200,
    "free_space": 385020329984,
    "created_on": "2014-11-20T19:37:22Z",
    "updated_on": "2015-03-09T13:36:05Z",
    "multiplehostaccess": 0,
    "location": "4e43dd32-c6b7543a-32bf-0010187f038c",
    "uncommitted": 845539212800,
    "ems_ref_obj": "--- !ruby/string:VimString\nstr: datastore-15624\nnxsiType: :ManagedObjectReference\nvimType: :Datastore\n",
    "directory_hierarchy_supported": true,
    "thin_provisioning_supported": true,

```

```

    "raw_disk_mappings_supported": true,
    "master": false,
    "ems_ref": "datastore-15624"
  }
  "software": [
    {
      "href": "http://localhost:3000/api/vms/320/software/1",
      "id": "1",
      "name": "OpenOffice",
      "vendor": "OpenOffice.org",
      "vm_or_template_id": "166"
    }
  ]
}

```

As another example, one can query good details on hosts:

```

GET /api/hosts/8?
attributes=custom_attributes,ext_management_system,resource_pools,storages,vms,hardware

```

of course, one needs to be careful with queries like these as list of vms for a host could be quite large.

Virtual attributes can also be queried from one-to-one relationships via the dot notation as follows:

```

GET /api/hosts/8?
attributes=ext_management_system.id,ext_management_system.guid,ext_management_system.name

```

```

{
  "href": "http://localhost:3000/api/hosts/8",
  "id": "8",
  "name": "test1.sample.com",
  "hostname": "test1.sample.com",
  "ipaddress": "test1.sample.com",
  "vmm_vendor": "vmware",
  "vmm_version": "5.0.0",
  "vmm_product": "ESXi",
  "vmm_buildnumber": "515841",
  ...
  "ext_management_system": {
    "name": "vcenter50",
    "guid": "e84e8c58-bdbd-11e4-8983-b8e85646e742",
    "id": "6"
  }
}

```

With attributes, database attributes, virtual attributes, and relationships can be specified together as in the following example:

```

GET /api/vms/166?attributes=name,raw_power_state,ipaddresses,storage.name

```

```

{
  "href": "http://localhost:3000/api/vms/166",
  "id": "166",
  "name": "Dev Nightly Appl",
  "raw_power_state": "poweredOff",
  "ipaddresses": [
    "192.168.253.1"
  ],
  "storage": {
    "name": "StarM1-Dev"
  }
}

```


This is helpful when specific information is needed out of resources and helps with response time when querying large number of resources as in the following example:

```
GET /api/vms?limit=1000&offset=1000&expand=resources&
attributes=name,raw_power_state,ipaddresses,storage.name
```

- [Bulk Querying](#)

Bulk Querying

While the capability to fetch an individual resource via `/api/:collection/:id` is available, fetching multiple resources via a filtered search is possible but has limitation on GET url limits and is cumbersome on the filtering syntax.

The Bulk Queries provides this capability by implementing a *query* POST action on the primary collections.

```
POST /api/vms
```

```
{
  "action" : "query",
  "resources" : [
    { "guid" : "...." },
    { "guid" : "...." },
    ...
  ]
}
```

The *query* interface is flexible in that it allows resources to be specified by different relevant attributes. These include:

Identifying Attributes
href
id
guid (if applicable)

Some other attributes can also be used as identifying attributes as shown in the following table:

Collection	Optional Identifying Attributes
/api/groups	description
/api/roles	name
/api/tags	name
/api/users	name, userid

As with regular GETs, queries return expanded resources, but also support other query parameters as in the following example:

```
POST /api/vms?attributes=name,raw_power_state
```

```
{
  "action" : "query",
  "resources" : [
    { "id" : 10 },
    { "id" : 20 },
    { "id" : 23 },
    { "id" : 28 },
    { "id" : 30 }
  ]
}
```

Sample response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/vms/10",
      "id": "10",
      "name": "test_vm_1",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/20",
      "id": "20",
      "name": "demo_server_A",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/23",
      "id": "23",
      "name": "demo_server_B",
      "raw_power_state": "down"
    },
    {
      "href": "http://localhost:3000/api/vms/28",
      "id": "28",
      "name": "aab_dev_vm",
      "raw_power_state": "up"
    },
    {
      "href": "http://localhost:3000/api/vms/30",
      "id": "30",
      "name": "ipa_dev",
      "raw_power_state": "down"
    }
  ]
}
```

- [Collection Filtering](#)

Collection Filtering

When querying collections, filters can be specified by providing one or more `filter[]=...` parameter.

```
GET /api/:collection?filter[]=...&filter[]=...&filter[]=...
```

Syntax for a filter is as follows:

```
filter[]=attribute op value
```

By default, filters are logically *anded* together, for a logical or, prefix the attribute with *or* as follows:

```
filter[]=or attribute op value
```

Where op supports the following operators:

Operator	Description	Operands
=	Equality	Numeric or string. For String, supports % or * as wildcard NULL or nil can also be specified instead of quoted string
!=	Inequality	

Operator	Description	Operands
<	Less Than	Numeric
<=	Less Than or Equal	Numeric
>=	Greater Than or Equal	Numeric
>	Greater Than	Numeric

When strings, value to be quoted in single or double quotes. When datetimes, only strict comparison operators are supported.

Example Query of VMs named sample*, created_on > 2019-09-01, and return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor&filter[]=created_on>2019-09-01&filter[]=name='sample%'
```

Example Query looking for services that are retired but have an unspecified service_id

```
GET /api/services?
expand=resources&attributes=retired&filter[]=service_id=nil&filter[]=retired=true
```

Example Query of VMs with power_state of on or off:

```
GET /api/vms?expand=resources&filter[]=power_state=[on,off]
```

Alternately, you can use the following more verbose syntax using OR:

```
GET /api/vms?
expand=resources&filter[]=power_state=off&filter[]=or%20power_state=on
```

Example Query of VMs with more than 4 CPU sockets or more than 16,000 MB of RAM (OR):

```
GET /api/vms?expand=resources&filter[]=num_cpu>4&filter[]=or%20ram_size>16000
```

Example Query of VMs without a specific value (attribute negation):

```
GET /api/vms?expand=resources&filter[]=power_state!=unknown
```

Example Query of VMs where a field is NOT NULL (vms that are either retired or not):

```
GET /api/vms?expand=resources&filter[]=retired!=nil
GET /api/vms?expand=resources&filter[]=retired!=NULL
GET /api/vms?expand=resources&filter[]=retired!=null
```

Example Query of VMs using a boolean:

```
GET /api/vms?expand=resources&filter[]=retired=true
```

Example Query of VMs created before a specific time:

```
GET /api/vms?expand=resources&filter[]=created_on<2023-01-01
```

Attributes specified via the filter[] parameter can include virtual attributes, including one level association of the virtual attribute as follows:

The HTTP **OPTIONS** request can be used on a collection to display all of the collection specific metadata, including the available virtual attributes. For more information please refer to the [Collection Metadata](#) page.

```
GET /api/vms?
attributes=name,vendor,hardware&filter[]=vendor='vmware'&filter[]=hardware.memory_mb>=8192
```

Here filtering on the virtual attribute *hardware* including the one level association *memory_mb* off hardware.

- [Pagination](#)
 - [About max_results_per_page](#)

Pagination

Pagination is enabled by default. When querying collections and the *limit* query parameter is not specified, only 1000 resources are returned by default. This is driven by the new advanced setting:

```
:api:  
:max_results_per_page: 1000
```

Example GET with offset and limit specified.

```
GET /api/vms?offset=2&limit=2&sort_by=name&expand=resources
```

```
{  
  "name"      : "vms",  
  "count"     : 7,  
  "subcount"  : 2,  
  "pages"     : 4,  
  "resources" : [  
    {  
      "href"   : "http://localhost:3000/api/vms/7254",  
      "id"     : "7254",  
      "vendor" : "vmware",  
      "name"   : "test-vm1",  
      ...  
    },  
    {  
      "href"   : "http://localhost:3000/api/vms/7257",  
      "id"     : "7257",  
      "vendor" : "vmware",  
      "name"   : "test-vm2",  
      ...  
    }  
  ],  
  "actions" : [  
    ...  
  ],  
  "links" : {  
    "self"      : "http://localhost:3000/api/vms?offset=2&limit=2&sort_by=name&expand=resources",  
    "next"      : "http://localhost:3000/api/vms?offset=4&limit=2&sort_by=name&expand=resources",  
    "previous"  : "http://localhost:3000/api/vms?offset=0&limit=2&sort_by=name&expand=resources",  
    "first"     : "http://localhost:3000/api/vms?offset=0&limit=2&sort_by=name&expand=resources",  
    "last"      : "http://localhost:3000/api/vms?offset=6&limit=2&sort_by=name&expand=resources"  
  }  
}
```

With pagination enabled by default, collection queries now return the following:

- “pages” metadata which is the total number of pages for the *limit* specified (or the default 1000).
- “links” metadata section which includes the necessary URL’s for fetching the current page, next and previous pages, as well as the first and last pages.

Notes:

- When the page being returned is the first page (i.e. offset = 0), the *previous* link is not returned.
- When the page being returned is the last page, the *next* page link is not returned.
- For small result sets where *limit* is less than max_results_per_page, the *limit* will be honored.
- For large result sets where the *limit* is greater than max_results_per_page, *limit* will be ignored and the result will be limited to the first max_results_per_page resources.

About max_results_per_page

If the default value for max_results_per_page does not meet your needs, it can be changed using the Advanced Settings feature in the UI. However, it is *NOT* recommended to set it to some arbitrarily large number in an effort to circumvent the pagination feature. The pagination feature is designed to prevent excessive load on the API workers, which is easy to do with large collections such as metric rollups and event streams that may have millions of records.

Infrastructure Management API Reference

In this guide you will find a detailed reference for the Infrastructure Automation – Infrastructure Management REST API.

- [Primary Collections](#)
- [Actions](#)
- [Alert Management](#)
- [Authentication Management](#)
- [Automate Model](#)
- [Automate Domains](#)
- [Automation Requests](#)
- [Automate Workspaces](#)
- [Categories and Tags](#)
- [Chargebacks, Rates and Assignments](#)
- [Cloud Volumes](#)
- [Conditions](#)
- [Configuration Script Source Management](#)
- [Conversion Hosts](#)
- [Custom Actions](#)
- [Custom Attributes](#)
- [Custom Button Management](#)
- [Event Streams](#)
- [Flavor Management](#)
- [Generic Object Management](#)
- [Group Management](#)
- [Host Management](#)
- [Instance Management](#)
- [Metric Rollups](#)
- [Network Routers](#)
- [Notifications Support](#)
- [Orchestration Support](#)
- [Physical Server Management](#)
- [Picture Management](#)
- [Policy Management](#)

- [Policy Collections](#)
- [Provider Support](#)
- [Provision Requests](#)
- [Request Support](#)
- [Reports Support](#)
- [Role Management](#)
- [Security Groups](#)
- [Service Management](#)
- [Service Queries](#)
- [Service Custom Actions](#)
- [Service Dialogs](#)
- [Service Orders](#)
- [Setting Ownership](#)
- [Settings Management](#)
- [Snapshot Management](#)
- [Tags and Tagging](#)
- [Tasks](#)
- [Tenant Management](#)
- [Transformation Mappings](#)
- [User Management](#)
- [Virtual Machine Management](#)
- [Collections](#)

Collections

The following tables shows the currently available primary collections accessible via:

GET /api/:collection

Where :collection is one of the following:

Collection	Description
actions	Actions
alert_definition_profiles	Alert Definition Profiles
alert_definitions	Alert Definitions
alerts	Alerts
auth_key_pairs	Auth Key Pairs
authentications	Authentications
automate	Automate Model
automate_classes	Automate Classes
automate_domains	Automate Domains
automate_workspaces	Automate Workspaces
automation_requests	Automation Requests
availability_zones	Availability Zones
categories	Categories
chargebacks	Chargebacks
cloud_networks	Cloud Networks
cloud_object_store_containers	Cloud Object Store Containers
cloud_subnets	Cloud Subnets

Collection	Description
cloud_templates	Cloud Templates
cloud_tenants	Cloud Tenants
cloud_volume_types	Cloud Volume Types
cloud_volumes	Cloud Volumes
clusters	Clusters
conditions	Conditions
configuration_profiles	Configuration Profiles
configuration_script_payloads	Configuration Script Payloads
configuration_script_sources	Configuration Script Source
configuration_scripts	Configuration Scripts
configured_systems	Configured Systems
container_groups	Container Groups
container_images	Container Images
container_nodes	Container Nodes
container_projects	Container Projects
container_templates	Container Templates
container_volumes	Container Volumes
containers	Containers
conversion_hosts	Conversion Hosts
currencies	Currencies
custom_button_sets	Custom Button Sets
custom_buttons	Custom Buttons
customization_scripts	Customization Scripts
customization_templates	Customization Templates
data_stores	Datastores
enterprises	Enterprises
event_streams	Event Streams
events	Events
features	Product Features
firmware_registries	Firmware Registries
firmwares	Firmwares
flavors	Flavors
floating_ips	Floating IPs
generic_object_definitions	Generic Object Definitions
generic_objects	Generic Objects
groups	Groups
guest_devices	Guest Devices
hosts	Hosts
instances	Instances
lans	Lans
load_balancers	Load Balancers
measures	Measures
metric_rollups	Metric Rollups
network_routers	Network Routers

Collection	Description
notifications	User's past notifications
orchestration_stacks	Orchestration Stacks
orchestration_templates	Orchestration Template
physical_chassis	Physical Chassis
physical_racks	Physical Racks
physical_servers	Physical Servers
physical_storages	Physical Storages
physical_switches	Physical Switches
pictures	Pictures
policies	Policies
policy_actions	Actions
policy_profiles	Policy Profiles
providers	Providers
provision_dialogs	Provisioning Dialogs
provision_requests	Provision Requests
pxe_images	PXE Images
pxe_menus	PXE Menus
pxe_servers	PXE Servers
rates	Chargeback Rates
regions	Regions
reports	Reports
request_tasks	Request Tasks
requests	Requests
resource_pools	Resource Pools
results	Report Results
roles	Roles
search_filters	Search Filters
security_groups	Security Groups
servers	EVM Servers
service_catalogs	Service Catalogs
service_dialogs	Service Dialogs
service_offerings	Service Offerings
service_orders	Service Orders
service_parameters_sets	Service Parameters Sets
service_requests	Service Requests
service_templates	Service Templates
services	Services
settings	Settings
switches	Switches
tags	Tags
tasks	Tasks
templates	Templates
tenant_groups	Tenant Groups
tenants	Tenants

Collection	Description
transformation_mappings	Transformation Mappings
users	Users
vms	Virtual Machines
widgets	Miq Widgets
zones	Zones

- [Actions](#)
 - [Querying Actions](#)
 - [Creating Actions](#)
 - [Editing Actions](#)
 - [Deleting Actions](#)

Actions

Management of actions is provided via the following collection:

/api/actions

Full CRUD actions on resources is available:

- [Querying Actions](#)
- [Creating Actions](#)
- [Editing Actions](#)
- [Deleting Actions](#)

Querying Actions

Querying all actions in the system is simply:

GET /api/actions

Getting details on a specific Action:

GET /api/actions/:id

Creating Actions

Actions can be created via a POST to the actions collection or via the **create** action signature which also allows creation of multiple actions in a single request.

POST /api/actions

```
{
  "name" : "sample_action",
  "description" : "Sample Action",
  "action_type" : "custom_automation",
  "options" : {
    "ae_message" : "message",
    "ae_request" : "request",
    "ae_hash" : { "sample_key" : "sample_value" }
  }
}
```

or creating multiple actions:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "action_1", "description" : "Action 1", ... },
    { "name" : "action_2", "description" : "Action 2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating actions.

Editing Actions

POST /api/actions/:id

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Description"
  }
}
```

or editing multiple actions:

POST /api/actions

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/actions/101",
      "description" : "Updated Description 1"
    },
    {
      "href" : "http://localhost:3000/api/actions/102",
      "description" : "Updated Description 2"
    },
    ...
  ]
}
```

Deleting Actions

Actions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/actions/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/actions/101

Deleting multiple actions can be done as follows:

POST /api/actions

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/actions/101" },
    { "href" : "http://localhost:3000/api/actions/102" },
    ...
  ]
}
```

- [Alert Management](#)
 - [Querying Alerts](#)
 - [Querying Alert Actions](#)
 - [Creating Alert Actions](#)
 - [Querying Alert Definitions](#)
 - [Creating Alert Definitions](#)
 - [Editing Alert Definitions](#)
 - [Deleting Alert Definitions](#)

Alert Management

Management of alerts, alert actions and alert definitions is provided via the following collections and subcollections:

```
/api/alerts
/api/alerts/:id/alert_actions
/api/alert_definitions
```

Queries of alerts:

- [Querying Alerts](#)

Queries and creation of alert actions:

- [Querying Alert Actions](#)
- [Creating Alert Actions](#)

Full CRUD actions on alert definitions:

- [Querying Alert Definitions](#)
- [Creating Alert Definitions](#)
- [Editing Alert Definitions](#)
- [Deleting Alert Definitions](#)

Querying Alerts

Querying all alerts in the system is simply:

```
GET /api/alerts
```

Getting details on a specific alert:

```
GET /api/alerts/:id
```

Querying Alert Actions

Querying all alert actions of a specific alert is done via the subcollection as follows:

```
GET /api/alerts/:alert_id/alert_actions
```

Querying an alert action of a specific alert is done via the subcollection resource as follows:

```
GET /api/alerts/:alert_id/alert_actions/:alert_action_id
```

Creating Alert Actions

Creating an alert action for a specific alert can be done by the following POST:

```
POST /api/alerts/:alert_id/alert_actions
{
  "action_type" : "custom_automation",
  "comment" : "Comment about the Alert Action"
}
```

Querying Alert Definitions

Querying all alert definitions in the system is simply:

```
GET /api/alert_definitions
```

Getting details on a specific alert definition:

```
GET /api/alert_definitions/:id
```

Creating Alert Definitions

Alert definitions can be created via a POST to the alert definitions collection or via the **create** action signature which also allows creation of multiple alert definitions in a single request.

```
POST /api/alert_definitions
{
  "description" : "Test Alert Definition",
  "db" : "ContainerNode",
  "expression" : { "eval_method" : "dwh_generic", "mode" : "internal", "options" :
{} },
  "options" : { "notifications" : { "delay_next_evaluation" : 600, "evm_event"
: {} } },
  "enabled" : true
}
```

or creating multiple alert definitions:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "Sample Alert Definition 1", ... },
    { "description" : "Sample Alert Definition 2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating alert definitions.

Editing Alert Definitions

POST /api/alert_definitions/:id

```
{
  "action" : "edit",
  "resource" : {
    "options" : { "notifications" : { "delay_next_evaluation" : 1200, "evm:event"
: {} } } }
}
```

or editing multiple alert definitions:

POST /api/alert_definitions

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/alert_definitions/101",
      "options" : { "notifications" : { "delay_next_evaluation" : 1200,
"evm:event" : {} } } }
    },
    {
      "href" : "http://localhost:3000/api/alert_definitions/102",
      "options" : { "notifications" : { "delay_next_evaluation" : 1200,
"evm:event" : {} } } }
    },
    ...
  ]
}
```

Deleting Alert Definitions

Alert definitions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/alert_definitions/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/alert_definitions/101

Deleting multiple alert definitions can be done as follows:

POST /api/alert_definitions

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/alert_definitions/101" },
    { "href" : "http://localhost:3000/api/alert_definitions/102" },
    ...
  ]
}
```

- [Authentications](#)
 - [Querying Authentications](#)
 - [Creating Authentications](#)

- [Editing Authentications](#)
- [Refreshing Authentications](#)
- [Deleting Authentications](#)

Authentications

Management of authentications is provided via the following collection:

`/api/authentications`

Full CRUD actions on authentications is available:

- [Querying Authentications](#)
- [Creating Authentications](#)
- [Editing Authentications](#)
- [Refreshing Authentications](#)
- [Deleting Authentications](#)

Querying Authentications

Querying all authentications in the system is simply:

`GET /api/authentications`

Getting details on a specific authentication:

`GET /api/authentications/:id`

One can also query authentications of a configuration script payload's manager as follows:

`GET /api/configuration_script_payloads/:id/authentications`

Or getting details on a specific authentication:

`GET /api/configuration_script_payloads/:id/authentications/:authentication_id`

Creating Authentications

Authentications can be created via a POST to the authentications collection or via the **create** action signature which also allows creation of multiple authentications in a single request.

`POST /api/authentications`

```
{
  "description" : "Authentication Description",
  "name" : "SomeCredentials",
  "related" : {},
  "type" : "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential",
  "manager_resource" : { "href" : "http://localhost:3000/api/providers/7" }
}
```

or creating multiple authentications:

```
{
  "action" : "create",
  "resources" : [
```

```

    { "description" : "System Credentials", "name" : "SystemCreds", ... },
    { "description" : "Admin Credentials", "name" : "AdminCreds", ... },
    ...
  ]
}

```

Optionally, authentications can be created for a provider of a specific configuration script payload, thus eliminating the need of specifying a provider.

POST /api/configuration_script_payloads/:id/authentications

```

{
  "description" : "Authentication Description",
  "name" : "SomeCredentials",
  "related" : {},
  "type" : "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential"
}

```

Also supporting bulk creates:

```

{
  "action" : "create",
  "resources" : [
    { "description" : "System Credentials", "name" : "SystemCreds", ... },
    { "description" : "Admin Credentials", "name" : "AdminCreds", ... },
    ...
  ]
}

```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating authentications.

Editing Authentications

POST /api/authentications/:id

```

{
  "action" : "edit",
  "resource" : {
    "name" : "UpdatedCredentials"
  }
}

```

or editing multiple authentications:

POST /api/authentications

```

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/authentications/101",
      "description" : "Updated Sample Credentials 1",
      "name" : "UpdatedCredentials1"
    },
    {
      "href" : "http://localhost:3000/api/authentications/102",
      "description" : "Updated Sample Credentials 2",
      "name" : "UpdatedCredentials2"
    },
    ...
  ]
}

```

```
]
}
```

Refreshing Authentications

Authentications can be refreshed by posting a **refresh** action to a single resource or to multiple resources in bulk by targeting the collection.

POST /api/authentications/:id

```
{
  "action" : "refresh"
}
```

or refreshing multiple authentications:

POST /api/authentications

```
{
  "action" : "refresh",
  "resources" : [
    { "id" : "51" },
    { "id" : "52" }
  ]
}
```

Example:

POST /api/authentications/51

```
{
  "action" : "refresh"
}
```

Response:

```
{
  "success" : true,
  "message" : "Refreshing Authentication id:51 name:'SampleCredentials'",
  "task_id" : "8",
  "task_href" : "http://localhost:3000/api/tasks/8",
  "tasks" : [
    {
      "id" : "8",
      "href" : "http://localhost:3000/api/tasks/8"
    }
  ]
}
```

Deleting Authentications

Authentications can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/authentications/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/authentications/101

Deleting multiple authentications can be done as follows:

POST /api/authentications

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/authentications/101" },
    { "href" : "http://localhost:3000/api/authentications/102" },
    ...
  ]
}
```

- [Automate](#)
 - [Usage](#)
 - [Querying Domains](#)
 - [Querying By Fully Qualified Name](#)
 - [Querying First Level Children Nodes](#)
 - [Querying Subtree](#)
 - [Querying State Machines Subtree](#)

Automate

Querying the Automate model is provided via the following endpoint:

/api/automate

- [Usage](#)
- [Querying Domains](#)
- [Querying By Fully Qualified Name](#)
- [Querying First Level Children Nodes](#)
- [Querying Subtree](#)
- [Querying State Machines Subtree](#)

Usage

Querying the Automate Model via /api/automate allows queries of:

- top level Automate Domains
- any object in the Automate Model
- arbitrary depth via the **depth** parameter

depth value	data returned
0	Just node itself
1	Node itself plus its first level children nodes
#	Node itself plus # levels of children nodes
-1	Node itself plus the whole subtree underneath it

- Automate model object tree to include state machine endpoints via the **search_options** parameter with value **state_machines**

Example queries:

```
GET /api/automate
GET /api/automate/manageiq
GET /api/automate/manageiq?depth=2
GET /api/automate/manageiq?depth=-1&search_options=state_machines
```

Querying Domains

Querying the Automate Domains can be done via GET on the /api/automate endpoint as follows:

```
GET /api/automate
```

```
{
  "name": "automate",
  "subcount": 4,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain",
      "id": "2352",
      "name": "ManageIQ",
      "created_on": "2016-10-31T14:58:09Z",
      "updated_on": "2016-10-31T14:58:09Z",
      "description": "Base domain",
      "updated_by": "system",
      "priority": 0,
      "enabled": true,
      "tenant_id": "1",
      "source": "system"
    },
    ...
  ]
}
```

Note:

The fqname is always returned by this query

If desired a sub-set of attributes can be asked via as follows:

```
GET /api/automate?attributes=name,source
```

```
{
  "name": "automate",
  "subcount": 4,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "name": "ManageIQ",
      "source": "system"
    },
    ...
  ]
}
```

Querying By Fully Qualified Name

Querying any object in the Automate Model by its fully qualified name as follows:

Note:

The fqname in the query is *not* case sensitive

```
GET /api/automate/manageiq
```

```
{
  "name": "automate",
  "subcount": 1,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain",
      "id": "2352",
      "name": "ManageIQ",
      "created_on": "2016-10-31T14:58:09Z",
      "updated_on": "2016-10-31T14:58:09Z",
      "description": "Base domain",
      "updated_by": "system",
      "priority": 0,
      "enabled": true,
      "tenant_id": "1",
      "source": "system"
    }
  ]
}
```

Another example showing fetching an automate class object:

```
GET /api/automate/manageiq/system/request
```

```
{
  "name": "automate",
  "subcount": 1,
  "resources": [
    {
      "fqname": "/ManageIQ/System/Request",
      "domain_fqname": "/System/Request",
      "klass": "MiqAeClass",
      "id": "75",
      "description": "Automation Requests",
      "name": "Request",
      "created_on": "2015-12-09T20:56:44Z",
      "updated_on": "2015-12-09T20:56:44Z",
      "namespace_id": "41",
      "updated_by": "system"
    }
  ]
}
```

Automate model nodes returned include both:

- **fqname**: the fully qualified name of the node which is rooted at /.
 - i.e. /ManageIQ/System/Request
- **domain_fqname**: the fully qualified name of the node rooted at the domain, which is **fqname** without the Domain prefix
 - i.e. /System/Request

Querying First Level Children Nodes

When specifying a depth of 1, the Automate model node and its first level children nodes are returned.

In this example, we are asking for the /ManageIQ Domain and its direct children nodes.

```
GET /api/automate/manageiq?depth=1&attributes=domain_fqname,klass
```

```
{
  "name": "automate",
  "subcount": 9,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain"
    },
    {
      "fqname": "/ManageIQ/System",
      "domain_fqname": "/System",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Service",
      "domain_fqname": "/Service",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Infrastructure",
      "domain_fqname": "/Infrastructure",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/GenericObject",
      "domain_fqname": "/GenericObject",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Deployment",
      "domain_fqname": "/Deployment",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Control",
      "domain_fqname": "/Control",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/ConfigurationManagement",
      "domain_fqname": "/ConfigurationManagement",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/Cloud",
      "domain_fqname": "/Cloud",
      "klass": "MiqAeNamespace"
    }
  ]
}
```

Querying Subtree

When specifying a depth of -1, the Automate model node and all of the nodes underneath it are returned.

```
GET /api/automate/manageiq?depth=-1&attributes=klass
```

```
{
  "name": "automate",
```

```

"subcount": 935,
"resources": [
  {
    "fqname": "/ManageIQ",
    "klass": "MiqAeDomain"
  },
  {
    "fqname": "/ManageIQ/System",
    "klass": "MiqAeNamespace"
  },
  {
    "fqname": "/ManageIQ/System/Event",
    "klass": "MiqAeNamespace"
  },
  {
    "fqname": "/ManageIQ/System/Event/RequestEvent",
    "klass": "MiqAeNamespace"
  },
  ...
]
}

```

Querying State Machines Subtree

In this example, we are querying the ManageIQ Automate Domain model tree but only asking for branches that include state machine nodes.

```

GET /api/automate/manageiq?
depth=-1&attributes=domain_fqname,klass&search_options=state_machines

```

```

{
  "name": "automate",
  "subcount": 104,
  "resources": [
    {
      "fqname": "/ManageIQ",
      "domain_fqname": "/",
      "klass": "MiqAeDomain"
    },
    {
      "fqname": "/ManageIQ/System",
      "domain_fqname": "/System",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods",
      "domain_fqname": "/System/CommonMethods",
      "klass": "MiqAeNamespace"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods/QuotaStateMachine",
      "domain_fqname": "/System/CommonMethods/QuotaStateMachine",
      "klass": "MiqAeClass"
    },
    {
      "fqname": "/ManageIQ/System/CommonMethods/QuotaStateMachine/quota",
      "domain_fqname": "/System/CommonMethods/QuotaStateMachine/quota",
      "klass": "MiqAeInstance"
    },
    ...
  ]
}

```

- [Automate Domains](#)
 - [Querying Domains](#)
 - [Accessing Domains](#)
 - [Refresh From Source](#)

Automate Domains

Querying the Automate Domains and triggering actions on domains is provided via the following endpoint:

`/api/automate_domains`

- [Querying Domains](#)
- [Accessing Domains](#)
- [Refresh From Source](#)

Querying Domains

Querying the Automate Domains can be done via GET on the `/api/automate_domains` endpoint and can be queried like any other collection:

`GET /api/automate_domains`

```
{
  "name": "automate_domains",
  "count": 9,
  "subcount": 9,
  "resources": [
    {
      "href": "http://localhost:3000/api/automate_domains/2352"
    },
    {
      "href": "http://localhost:3000/api/automate_domains/2347"
    },
    ...
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/automate_domains"
    },
    {
      "name": "refresh_from_source",
      "method": "post",
      "href": "http://localhost:3000/api/automate_domains"
    }
  ]
}
```

Accessing Domains

Accessing individual domains can be done via their numeric id or their name (*case sensitive*) as follows:

`GET /api/automate_domains/2352`

`GET /api/automate_domains/ManageIQ`

Refresh From Source

When Automate domains are imported from external sources, i.e. Git, the *refresh_from_source* will re-fetch the Automate domain from external source and refresh the automate domain. This will refetch the Domain from the same reference, i.e. branch or tag and name.

POST /api/automate_domains/SampleDomain

```
{
  "action": "refresh_from_source"
}
```

All refresh_from_source actions are done asynchronously and the related Task id and href are returned as follows:

```
{
  "success": true,
  "message": "Refreshing Automate Domain id:2347 name:'SampleDomain' from git repository",
  "task_id": "61",
  "task_href": "http://localhost:3000/api/tasks/61",
  "href": "http://localhost:3000/api/automate_domains/SampleDomain"
}
```

If the Domain needs to be refreshed from a different branch or tag and name, the following can be passed in with the *refresh_from_source* action:

```
{
  "action": "refresh_from_source",
  "resource": {
    "ref_type": "branch" | "tag",
    "ref": "other_branch_or_tag_name"
  }
}
```

- [Automation Requests](#)
 - [Creating Requests](#)
 - [Editing Requests](#)
 - [Approving or Denying Requests](#)
 - [Approving a Request](#)
 - [Denying a Request](#)
 - [Monitoring Requests](#)
 - [Response:](#)
 - [Editing Request Tasks](#)

Automation Requests

Automation requests are made available via the following endpoint:

/api/automation_requests

- [Creating Requests](#)
- [Editing Requests](#)
- [Approving or Denying Requests](#)
- [Monitoring Requests](#)
- [Editing Request Tasks](#)

Creating Requests

Either by specifying a *create* action or by posting the automation request directly to **/api/automation_requests**.

As an example:

```
{
  "uri_parts" : {
    "namespace" : "System",
    "class"      : "Request",
    "instance"   : "InspectME",
    "message"    : "create"
  },
  "parameters" : {
    "var1" : "value 1",
    "var2" : "value 2",
    "minimum_memory" : 2048
  },
  "requester" : {
    "auto_approve" : true
  }
}
```

Editing Requests

Updating requests can be performed by posting an *edit* action as follows:

POST /api/automation_requests/11

```
{
  "action" : "edit",
  "resource" : {
    "options" : {
      "attrs" : {
        "var1" : "updated value1",
        "var2" : "updated value2",
        "minimum_memory" : 4096
      }
    }
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/automation_requests/11",
  "id": "11",
  "options": {
    "message": "create",
    "namespace": "System",
    "class_name": "Request",
    "instance_name": "InspectME",
    "user_id": "1",
    "attrs": {
      "var1" : "updated value1",
      "var2" : "updated value2",
      "minimum_memory" : 4096
    }
  },
  "tenant_id": "1",
  "approval_state": "approved",
  "status": "Ok",
}
```



```

"request_type": "automation",
"description": "Automation Task",
"type": "AutomationRequest",
"created_on": "2018-01-08T13:48:00Z",
"updated_on": "2018-01-08T13:56:15Z",
"requester_id": "1",
"requester_name": "Administrator",
"request_state": "pending",
"message": "Automation Request - Request Created",
"userid": "admin",
"process": true
}

```

Approving or Denying Requests

If an automation request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the automation request resource.

Approving a Request

POST /api/automation_requests/:id

```

{
  "action" : "approve",
  "reason" : "reason for approval"
}

```

Denying a Request

POST /api/automation_requests/:id

```

{
  "action" : "deny",
  "reason" : "reason for denial"
}

```

Monitoring Requests

Once an automation request is created, the response will be the queryable automation request itself, i.e. /api/automation_request/:id

Response:

```

{
  "results": [
    {
      "id": "10",
      "description": "Automation Task",
      "approval_state": "approved",
      "type": "AutomationRequest",
      "created_on": "2015-04-14T18:12:12Z",
      "updated_on": "2015-04-14T18:12:12Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "automation",
      "request_state": "pending",
      "status": "Ok",
      "options": {
        "message": "create",
        "namespace": "System",
        "class_name": "Request",

```

```

    "instance_name": "InspectME",
    "user_id": "1",
    "attrs": {
      "var1": "value 1",
      "var2": "value 2",
      "minimum_memory": 2048,
      "userid": "admin"
    }
  },
  "userid": "admin"
}
]
}

```

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state.

Note:

The request tasks of an automation request can also be queried by expanding the request_tasks subcollection as follows:

```
GET /api/automation_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/automation_requests/:id/request_tasks?expand=resources
```

Editing Request Tasks

Request tasks for Automation Requests can be updated as follows:

```
POST /api/automation_requests/:id/request_tasks/:request_task_id
```

```

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "request_param_a" : "value_a",
      "request_param_b" : "value_b"
    }
  }
}

```

- [Automate Workspaces](#)
 - [Querying Automate Workspaces](#)
 - [Request:](#)
 - [Response:](#)
 - [Editing Automate Workspaces](#)
 - [Encrypting Automate Workspace Passwords](#)
 - [Request:](#)
 - [Response:](#)
 - [Decrypting Automate Workspace Passwords](#)
 - [Request:](#)
 - [Response:](#)

Automate Workspaces

Management of Automate Workspaces is provided with the following collection:

/api/automate_workspaces

- [Querying Automate Workspaces](#)
- [Editing Automate Workspaces](#)
- [Encrypting Automate Workspace Passwords](#)
- [Decrypting Automate Workspace Passwords](#)

Querying Automate Workspaces

It is not allowed to query the automate workspace collection, only individual automate workspaces can be fetched by guid.

GET /api/automate_workspaces/:guid

Example GET Request and Response:

Request:

GET /api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f

Response:

```
{
  "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f",
  "id": "2",
  "guid": "430a66a0-b564-4341-b9b2-85d6f467f26f",
  "user_id": "1",
  "tenant_id": "1",
  "input": {
    "objects": {
      "root": {
        "var1": "a",
        "var2": "b"
      }
    },
    "method_parameters": {
      "arg1": "first_argument"
    }
  },
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    },
    {
      "name": "encrypt",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    },
    {
      "name": "decrypt",
      "method": "post",
      "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f"
    }
  ]
}
```

```

    }
  ]
}

```

Editing Automate Workspaces

Updating Automate Workspaces can be done via the *edit* action as follows:

POST /api/automate_workspaces/:guid

```

{
  "action" : "edit",
  "resource" : {
    "objects" : {
      "root" : {
        "var3" : "c"
      }
    }
  }
}

```

Encrypting Automate Workspace Passwords

Encrypting passwords in the automate workspace can be done as follows:

Request:

POST /api/automate_workspaces/:guid

```

{
  "action" : "encrypt",
  "resource" : {
    "object" : "method_parameters",
    "attribute" : "arg2",
    "value" : "unencrypted_password"
  }
}

```

Response:

```

{
  "href": "http://localhost:3000/api/automate_workspaces/430a66a0-b564-4341-b9b2-85d6f467f26f",
  "id": "2",
  "guid": "430a66a0-b564-4341-b9b2-85d6f467f26f",
  "user_id": "1",
  "tenant_id": "1",
  "input": {
    "objects": {
      "root": {
        "var1": "a",
        "var2": "b"
      }
    },
    "method_parameters": {
      "arg1": "first_argument",
      "arg2": "password:*****"
    }
  }
}

```

Decrypting Automate Workspace Passwords

Encrypted passwords in the automate workspace can be decrypted individually as follows:

Request:

POST /api/automate_workspaces/:guid

```
{
  "action" : "decrypt",
  "resource" : {
    "object" : "method_parameters",
    "attribute" : "arg2"
  }
}
```

Response:

```
{
  "object" : "method_parameters",
  "attribute" : "arg2",
  "value" : "unencrypted_password"
}
```

- [Categories & Tags](#)
 - [Querying Categories](#)
 - [Querying Tags](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)
 - [Creating Categories](#)
 - [Response:](#)
 - [Creating Tags](#)
 - [Editing Categories & Tags](#)
 - [Deleting Categories & Tags](#)
 - [Request:](#)
 - [Response:](#)

Categories & Tags

Management of Categories and Tags is provided via the following collections:

`/api/categories`

`/api/tags`

Full CRUD actions on both categories and tags are made available.

- [Querying Categories](#)
- [Querying Tags](#)
- [Creating Categories](#)
- [Creating Tags](#)
- [Editing Categories & Tags](#)

- [Deleting Categories & Tags](#)

Querying Categories

Querying all categories in the system is simply:

```
GET /api/categories
```

Or getting attributes of interest like *name*, *description* or *ns* for the namespace:

```
GET /api/categories?expand=resources&attributes=ns,name,description
```

Querying Tags

Querying all tags in the system is simply:

```
GET /api/tags
```

Querying tags of a specific category is done as follows:

```
GET /api/categories/:id/tags
```

Example Query for Location Category:

Request:

```
GET /api/categories/1
```

Response:

```
{
  "href": "http://localhost:3000/api/categories/1",
  "id": "1",
  "description": "Location",
  "read_only": false,
  "syntax": "string",
  "single_value": true,
  "example_text": "The geographic location of the resource, such as New York,
Chicago, or London.",
  "tag_id": "4",
  "parent_id": "0",
  "show": true,
  "default": true,
  "name": "location",
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/categories/1"
    }
  ]
}
```

Querying all tags of the Location category:

Request:

```
GET /api/categories/1/tags?expand=resources&attributes=name
```

Response:

```
{
  "name": "tags",
  "count": 1389,
  "subcount": 4,
  "resources": [
    {
      "href": "http://localhost:3000/api/categories/1/tags/8",
      "id": "8",
      "name": "/managed/location/paris"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/7",
      "id": "7",
      "name": "/managed/location/london"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/6",
      "id": "6",
      "name": "/managed/location/chicago"
    },
    {
      "href": "http://localhost:3000/api/categories/1/tags/5",
      "id": "5",
      "name": "/managed/location/ny"
    }
  ],
  "actions": [
    {
      "name": "create",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1/tags"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/categories/1/tags"
    }
  ]
}
```

For additional details on tags, the **category** and **classification** attributes can be queried on tags as follows:

```
GET /api/tags/5?attributes=category,classification
```

Additionally, a compound attribute called **categorization** is provided to return key information from both **category** and **classification** of tags as follows:

```
GET /api/tags/5?attributes=categorization
```

```
{
  "href": "http://localhost:3000/api/tags/5",
  "id": "5",
  "name": "/managed/location/ny",
  "categorization": {
```

```

    "name": "ny",
    "description": "New York",
    "category": {
      "name": "location",
      "description": "Location"
    },
    "display_name": "Location: New York"
  }
}

```

Creating Categories

Creating Categories can be done by posting the Category JSON to the categories collection directly, or via the **create** action signature as follows:

POST /api/categories

```

{
  "name" : "test",
  "description" : "Test Category"
}

```

or via the **create** action:

```

{
  "action" : "create",
  "resource" : {
    "name" : "test",
    "description" : "Test Category"
  }
}

```

Response:

```

{
  "results": [
    {
      "id": "1430",
      "description": "Test Category",
      "read_only": false,
      "syntax": "string",
      "single_value": false,
      "tag_id": "1439",
      "parent_id": "0",
      "show": true
    }
  ]
}

```

Attributes which can also be specified when creating categories include:

example_text
show
single_value
syntax

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Categories.

Creating Tags

New tags for a category can be created either as a tag collection post or via a tag subcollection **create** action to a specific category.

POST /api/tags

```
{
  "name" : "test_tag",
  "description" : "Test Tag",
  "category" : { "href" : "http://localhost:3000/api/categories/1430" }
}
```

The **category** above can identify a category by specifying either one of the **href**, **id** or **name** attributes.

Tags can optionally be created via the **create** action on the tags subcollection of a category as follows:

POST /api/categories/1430/tags

```
{
  "action" : "create",
  "resource" : {
    "name" : "test_tag",
    "description" : "Test Tag"
  }
}
```

or simply:

POST /api/categories/1430/tags

```
{
  "name" : "test_tag",
  "description" : "Test Tag"
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Tags.

Editing Categories & Tags

Editing Categories and Tags can be done via the **edit** action:

Example editing a category:

POST /api/categories/1430

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Category Description"
  }
}
```

Example editing a tag:

POST /api/tags/1441

```
{
  "action" : "edit",
  "resource" : {
    "name" : "updated_test_tag"
  }
}
```

```
}  
}
```

Deleting Categories & Tags

Deleting Categories and Tags can be done via either the **delete** post action or the DELETE HTTP method.

POST /api/categories/1430

```
{  
  "action" : "delete"  
}
```

or simply:

DELETE /api/categories/1430

Deleting associated tag example:

POST /api/tags/1441

```
{  
  "action" : "delete"  
}
```

or simply:

DELETE /api/tags/1441

One can also delete the tag when accessed via the subcollection

Request:

POST /api/categories/1430/tags

```
{  
  "action" : "delete",  
  "resources" : [  
    { "id" : "1441" },  
    { "id" : "1442" }  
  ]  
}
```

Response:

```
{  
  "results": [  
    {  
      "success": true,  
      "message": "tags id: 1441 deleting"  
    },  
    {  
      "success": true,  
      "message": "tags id: 1442 deleting"  
    }  
  ]  
}
```

Tags to be deleted can be specified via the **href**, **id** or **name** attribute.

- [Chargebacks, Rates & Assignments](#)
 - [Querying Chargebacks](#)

- [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)
- [Creating Chargeback Rates](#)
 - [Request:](#)
 - [Response:](#)
- [Updating Chargeback Rates](#)
 - [Request:](#)
 - [Response:](#)
- [Deleting Chargeback Rates](#)
 - [Request:](#)
 - [Response:](#)
- [Querying Assigned Chargeback Rates](#)
 - [Assigned chargeback rates on single rate:](#)
 - [Response:](#)
 - [Assigned chargeback rates on collection:](#)
 - [Response:](#)
- [Assign action for Chargeback Rates](#)
- [“Compute” Rate Type](#)
- [“Storage” Rate Type](#)
 - [Response:](#)
- [Basic Template for single chargeback request](#)
- [Basic Template for chargebacks collection request](#)
- [Enterprise](#)
- [Selected Providers](#)
- [Selected Clusters](#)
- [Tagged VMs and Instances](#)
- [Tagged Container Images](#)
- [Labeled Container Images](#)
- [Tenants](#)
- [Enterprise](#)
- [Selected Datastores](#)
- [Tagged Datastores](#)
- [Tenants](#)
- [Unassign action for Chargeback Rates](#)
- [Example of Chargeback Flow](#)
- [Create Chargeback Report](#)
 - [Chargeback Report Options](#)
- [Filtering according to resource](#)
 - [Chargeback for VMs \(db=ChargebackVm\)](#)
 - [Chargeback for Projects \(db=ChargebackContainerProject\)](#)
 - [Chargeback for Container Images \(db=ChargebackContainerImage\)](#)
- [Grouping Report Options](#)
 - [Chargeback for VMs \(db=ChargebackVm\)](#)
 - [Chargeback for Projects \(db=ChargebackContainerProject\)](#)
 - [Chargeback for Container Images \(db=ChargebackContainerImage\)](#)
 - [Example of request to create chargeback report](#)

- [Create Chargeback Rates](#)
- [Create Chargebacks \(top-level rate entity\)](#)
- [Assign Chargeback Rate](#)
- [Run Chargeback Report](#)
- [Results of Chargeback Report](#)

Chargebacks, Rates & Assignments

Queries of Chargebacks and Management of Rates is provided via the following collections:

`/api/chargebacks`

`/api/rates`

- [Querying Chargebacks](#)
- [Creating Chargeback Rates](#)
- [Updating Chargeback Rates](#)
- [Deleting Chargeback Rates](#)
- [Querying Assigned Chargeback Rates](#)
- [Assign action for Chargeback Rates](#)
- [Unassign action for Chargeback Rates](#)
- [Example of Chargeback Flow](#)

Querying Chargebacks

Querying all chargebacks in the system is simply:

GET `/api/chargebacks`

or Query a specific chargeback, for example:

GET `/api/chargebacks/1`

```
{
  "href": "http://localhost:3000/api/chargebacks/1",
  "id": "1",
  "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
  "description": "Default",
  "rate_type": "Compute",
  "created_on": "2014-11-20T19:10:03Z",
  "updated_on": "2014-11-20T19:10:03Z",
  "default": true
}
```

Optionally, one can query the rates details for a chargeback as follows:

GET `/api/chargebacks/:id/rates`

Example getting additional rates details(attributes)

Request:

GET /api/chargebacks/1/rates?

expand=resources&attributes=description,metric,group,source,friendly_rate

Response:

```
{
  "name": "rates",
  "count": 10,
  "subcount": 8,
  "resources": [
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/8",
      "id": "8",
      "description": "Fixed Compute Cost 2",
      "group": "fixed",
      "source": "compute_2",
      "friendly_rate": "0.0 Hourly"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/7",
      "id": "7",
      "description": "Fixed Compute Cost 1",
      "group": "fixed",
      "source": "compute_1",
      "friendly_rate": "1.0 Hourly"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/6",
      "id": "6",
      "description": "Used Disk I/O in KBps",
      "group": "disk_io",
      "metric": "disk_usage_rate_average",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 1.0 per Kbps from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/5",
      "id": "5",
      "description": "Used Network I/O in KBps",
      "group": "net_io",
      "metric": "net_usage_rate_average",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 0.0 per Kbps from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/4",
      "id": "4",
      "description": "Allocated Memory in MB",
      "group": "memory",
      "metric": "derived_memory_available",
      "source": "allocated",
      "friendly_rate": "Hourly @ 0.0 + 1.0 per Megabytes from 0.0 to Infinity"
    },
    {
      "href": "http://localhost:3000/api/chargebacks/1/rates/3",
      "id": "3",
      "description": "Used Memory in MB",
      "group": "memory",
      "metric": "derived_memory_used",
      "source": "used",
      "friendly_rate": "Hourly @ 0.0 + 0.0 per Megabytes from 0.0 to Infinity"
    },
    {

```

```

    "href": "http://localhost:3000/api/chargebacks/1/rates/2",
    "id": "2",
    "description": "Allocated CPU Count",
    "group": "cpu",
    "metric": "derived_vm_numvcpus",
    "source": "allocated",
    "friendly_rate": "Hourly @ 0.0 + 1.0 per Cpu from 0.0 to Infinity"
  },
  {
    "href": "http://localhost:3000/api/chargebacks/1/rates/1",
    "id": "1",
    "description": "Used CPU in MHz",
    "group": "cpu",
    "metric": "cpu_usagemhz_rate_average",
    "source": "used",
    "friendly_rate": "Hourly @ 0.0 + 0.0 per Megahertz from 0.0 to Infinity"
  }
]
}

```

Example getting additional of rates details(relations)

Request:

GET /api/rates?

expand=resources&attributes=chargeback_rate,detail_measure,detail_currency,chargeable_field,chargeback_tiers

Response:

```

{
  "name": "rates",
  "count": 138,
  "subcount": 138,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3090/api/rates/544",
      "id": "544",
      "enabled": true,
      "description": "Allocated CPU Count",
      "group": "cpu",
      "source": null,
      "metric": "derived_vm_numvcpus",
      "per_time": "hourly",
      "per_unit": "cpu",
      "friendly_rate": "Hourly @ 1.0 + 0.0 per Cpu from 0.0 to Infinity",
      "chargeback_rate_id": "8",
      "created_on": "2018-06-01T11:24:52Z",
      "updated_on": "2018-06-01T11:24:52Z",
      "chargeback_rate_detail_measure_id": null,
      "chargeback_rate_detail_currency_id": "1",
      "chargeable_field_id": "3",
      "sub_metric": null,
      "chargeback_rate": {
        "id": "8",
        "guid": "52a70f67-7ab8-47a0-ba36-7808ac2449e0",
        "description": "solinux1",
        "rate_type": "Compute",
        "created_on": "2018-06-01T11:24:52Z",
        "updated_on": "2018-06-01T11:24:52Z",
        "default": false
      }
    }
  ]
}

```

```

    },
    "detail_currency": {
      "id": "1",
      "code": "USD",
      "name": "United States Dollar",
      "full_name": "United States Dollar",
      "symbol": "$",
      "unicode_hex": "36",
      "created_at": "2019-06-13T18:10:32Z",
      "updated_at": "2019-07-09T13:42:36Z"
    },
    "chargeable_field": {
      "id": "3",
      "chargeback_rate_detail_measure_id": null,
      "metric": "derived_vm_numvcpus",
      "group": "cpu",
      "source": "allocated",
      "description": "Allocated CPU Count"
    },
    "chargeback_tiers": [
      {
        "id": "11010",
        "chargeback_rate_detail_id": "544",
        "start": 0.0,
        "finish": null,
        "fixed_rate": 1.0,
        "variable_rate": 0.0
      }
    ],
    "actions": [
      {
        "name": "edit",
        "method": "post",
        "href": "http://localhost:3090/api/rates/544"
      },
      {
        "name": "edit",
        "method": "patch",
        "href": "http://localhost:3090/api/rates/544"
      },
      {
        "name": "edit",
        "method": "put",
        "href": "http://localhost:3090/api/rates/544"
      },
      {
        "name": "delete",
        "method": "post",
        "href": "http://localhost:3090/api/rates/544"
      },
      {
        "name": "delete",
        "method": "delete",
        "href": "http://localhost:3090/api/rates/544"
      }
    ]
  },
  {
    "href": "http://localhost:3090/api/rates/510",
    "id": "510",
    "enabled": true,
    "description": "Allocated CPU Count",
    "group": "cpu",
    "source": null,

```

```

    "metric": null,
    "per_time": "hourly",
    "per_unit": "cpu",
    "friendly_rate": "Hourly @ 1.0 + 0.0 per Cpu from 0.0 to Infinity",
    "chargeback_rate_id": "7",
    "created_on": "2018-03-28T10:00:13Z",
    "updated_on": "2018-03-28T10:00:13Z",
    "chargeback_rate_detail_measure_id": null,
    "chargeback_rate_detail_currency_id": "3",
    "chargeable_field_id": "3",
    "sub_metric": null,
    "chargeback_rate": {
      "id": "7",
      "guid": "a72fd416-26ec-4aa0-bc94-0b741379dfcd",
      "description": "teamA_compute_all",
      "rate_type": "Compute",
      "created_on": "2018-03-28T10:00:13Z",
      "updated_on": "2018-03-28T10:00:13Z",
      "default": false
    },
    "detail_currency": {
      "id": "3",
      "code": "GBP",
      "name": "British Pound",
      "full_name": "British Pound",
      "symbol": "£",
      "unicode_hex": "163",
      "created_at": "2019-06-13T18:10:32Z",
      "updated_at": "2019-07-09T13:42:35Z"
    },
    "chargeable_field": {
      "id": "3",
      "chargeback_rate_detail_measure_id": null,
      "metric": "derived_vm_numvcpus",
      "group": "cpu",
      "source": "allocated",
      "description": "Allocated CPU Count"
    },
    "chargeback_tiers": [
      {
        "id": "10115",
        "chargeback_rate_detail_id": "510",
        "start": 0.0,
        "finish": 'infinity',
        "fixed_rate": 1.0,
        "variable_rate": 0.0
      }
    ]
  },
  {
    "href": "http://localhost:3090/api/rates/1859",
    "id": "1859",
    "enabled": true,
    "description": "My CPU allocation rate",
    "group": "cpu",
    "source": "allocated",
    "metric": null,
    "per_time": "daily",
    "per_unit": "megahertz",
    "friendly_rate": "Daily @ 2.0 + 0.0 per Megahertz from 0.0 to
1.0\nDaily @ 2.0 + 0.0 per Megahertz from 1.0 to Infinity",
    "chargeback_rate_id": "1",
    "created_on": "2020-05-26T10:34:39Z",
    "updated_on": "2020-05-26T10:34:39Z",

```



```

"chargeback_rate_detail_measure_id": "1",
"chargeback_rate_detail_currency_id": "5",
"chargeable_field_id": "3",
"sub_metric": null,
"chargeback_rate": {
  "id": "1",
  "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
  "description": "Default",
  "rate_type": "Compute",
  "created_on": "2020-04-23T14:43:50Z",
  "updated_on": "2020-04-23T15:00:29Z",
  "default": true
},
"detail_measure": {
  "id": "1",
  "name": "Bytes Units",
  "units": [
    "bytes",
    "kilobytes",
    "megabytes",
    "gigabytes",
    "terabytes"
  ],
  "units_display": [
    "B",
    "KB",
    "MB",
    "GB",
    "TB"
  ],
  "step": 1024,
  "created_at": "2020-04-23T14:43:50Z",
  "updated_at": "2020-04-23T15:00:28Z"
},
"detail_currency": {
  "id": "5",
  "code": "AUD",
  "name": "Australian Dollar",
  "full_name": "Australian Dollar",
  "symbol": "$",
  "unicode_hex": null,
  "created_at": "2019-06-13T18:10:32Z",
  "updated_at": "2019-07-09T13:42:35Z"
},
"chargeable_field": {
  "id": "3",
  "chargeback_rate_detail_measure_id": null,
  "metric": "derived_vm_numvcpus",
  "group": "cpu",
  "source": "allocated",
  "description": "Allocated CPU Count"
},
"chargeback_tiers": [
  {
    "id": "13114",
    "chargeback_rate_detail_id": "1859",
    "start": 0.0,
    "finish": 1.0,
    "fixed_rate": 2.0,
    "variable_rate": 0.0
  },
  {
    "id": "13115",
    "chargeback_rate_detail_id": "1859",

```

```

        "start": 1.0,
        "finish": 'infinity',
        "fixed_rate": 2.0,
        "variable_rate": 0.0
    }
]
},
...

```

Creating Chargeback Rates

Creating a Chargeback Rate is done by posting a new resource or *create* action to the rates collection.

Example follows:

Request:

POST /api/rates

```

{
  "per_time" : "daily",
  "chargeback_rate_id" : "1",
  "chargeable_field_id" : "1",
  "description": "My CPU allocation rate",
  "group" : "cpu",
  "per_unit" : "megahertz",
  "source" : "allocated",
  "chargeback_tiers": [
    {
      "start": 0.0,
      "finish": "infinity",
      "fixed_rate": 2.0,
      "variable_rate": 0.0
    }
  ]
}

```

Response:

```

{
  "results": [
    {
      "href": "http://localhost:3090/api/rates/1866",
      "id": "1866",
      "enabled": true,
      "description": "My CPU allocation rate",
      "group": "cpu",
      "source": "allocated",
      "metric": null,
      "per_time": "daily",
      "per_unit": "megahertz",
      "friendly_rate": "Daily @ 2.0 + 0.0 per MHz from 0.0 to Infinity",
      "chargeback_rate_id": "1",
      "created_on": "2020-05-28T13:28:15Z",
      "updated_on": "2020-05-28T13:28:15Z",
      "chargeback_rate_detail_measure_id": null,
      "chargeback_rate_detail_currency_id": null,
      "chargeable_field_id": "1",
      "sub_metric": null
    }
  ]
}

```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Chargeback Rates.

Updating Chargeback Rates

Updating rates can be done by posting **edit** actions on the rates resource.

Request:

```
POST /api/rates/16
```

```
{
  "action" : "edit",
  "resource" : { "description" : "Allocated NICs", "per_time": "hourly" }
}
```

Response:

```
{
  "href": "http://localhost:3090/api/rates/16",
  "id": "16",
  "description": "Allocated NICs",
  "per_time": "hourly",
  "chargeback_rate_id": "1",
  "chargeable_field_id": "3",
  "enabled": true,
  "group": "net_io",
  "source": null,
  "metric": null,
  "per_unit": "cpu",
  "friendly_rate": "Hourly @ 2.0 + 0.0 per Cpu from 0.0 to Infinity",
  "created_on": "2020-04-23T15:00:29Z",
  "updated_on": "2020-05-28T13:24:33Z",
  "chargeback_rate_detail_measure_id": "1",
  "chargeback_rate_detail_currency_id": "5",
  "sub_metric": null
}
```

Deleting Chargeback Rates

Deleting Chargeback Rates can be done via either the **delete** post action or the DELETE HTTP method.

Request:

```
POST /api/rates/16
```

```
{
  "action" : "delete"
}
```

Response:

```
{
  "success": true,
  "message": "rates id: 16 deleting",
  "href": "http://localhost:3000/api/rates/16"
}
```

or simply:

DELETE /api/rates/16

Querying Assigned Chargeback Rates

Assigned chargeback rates on single rate:

GET /api/chargebacks/16?attributes=assigned_to

Response:

```
{
  "href": "http://localhost:3090/api/chargebacks/2",
  "id": "2",
  "guid": "7d7aaf20-5214-11df-a888-001d09066d98",
  "description": "Default",
  "rate_type": "Storage",
  "created_on": "2019-08-07T16:48:04Z",
  "updated_on": "2020-04-20T13:27:07Z",
  "default": true,
  "assigned_to": [
    {
      "tag": {
        "href": "http://localhost:3090/api/tags/48",
        "name": "finance",
        "category": "department"
      }
    },
    {
      "tag": {
        "href": "http://localhost:3090/api/tags/26",
        "name": "test",
        "category": "environment"
      }
    }
  ]
}
```

Assigned chargeback rates on collection:

GET /api/chargebacks?expand=resources&attributes=assigned_to

Response:

```
{
  "name": "chargebacks",
  "count": 10,
  "subcount": 2,
  "subquery_count": 2,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3090/api/chargebacks/1",
      "id": "1",
      "guid": "b47a0ef0-4335-11df-aba8-001d09066d98",
      "description": "Default",
      "rate_type": "Compute",
      "created_on": "2019-08-07T16:48:04Z",
      "updated_on": "2020-05-13T09:36:21Z",
      "default": true,
      "assigned_to": [
        {

```

```

        "object": {
            "id": "1",
            "name": "Enterprise",
            "description": "Enterprise Root Object",
            "created_on": "2017-10-06T16:35:32Z",
            "updated_on": "2017-10-06T16:35:32Z",
            "settings": null
        }
    ],
    },
    {
        "href": "http://localhost:3090/api/chargebacks/2",
        "id": "2",
        "guid": "7d7aaf20-5214-11df-a888-001d09066d98",
        "description": "Default",
        "rate_type": "Storage",
        "created_on": "2019-08-07T16:48:04Z",
        "updated_on": "2020-05-13T09:36:21Z",
        "default": true,
        "assigned_to": [
            {
                "object": {
                    "id": "31",
                    "type":
"ManageIQ::Providers::Vmware::InfraManager::Storage",
                    ...
                }
            },
            {
                "object": {
                    "id": "19",
                    "type":
"ManageIQ::Providers::Vmware::InfraManager::Storage",
                    ...
                }
            },
            {
                "object": {
                    "id": "27",
                    "type":
"ManageIQ::Providers::Vmware::InfraManager::Storage",
                    ...
                }
            },
            {
                "object": {
                    "id": "14",
                    "type":
"ManageIQ::Providers::Vmware::InfraManager::Storage",
                    ...
                }
            }
        ]
    }
]
}

```

Assign action for Chargeback Rates

Assignments could be done for two rate types: “Compute” and “Storage”. There are few assignments type under each rates type:

“Compute” Rate Type

- [Enterprise](#)
- [Selected Providers](#)
- [Selected Clusters](#)
- [Tagged VMs and Instances](#)
- [Tagged Container Images](#)
- [Labeled Container Images](#)
- [Tenants](#)

“Storage” Rate Type

- [Enterprise](#)
- [Selected Datastores](#)
- [Tagged Datastores](#)
- [Tenants](#)

Example (chargebacks collection) - assign storage to chargeback rate with id 2 :

```
{
  "action" : "assign",
  "assignments" : [
    {
      "chargeback": {
        {
          "id" : 2
        },
      "resource": {
        {
          "href": "http://localhost:3090/api/data_stores/12"
        }
      }
    }
  ]
}
```

Response:

```
{
  "success": true,
  "message": "Rates assigned successfully",
  "result": [
    {
      "chargeback": {
        "href": "http://localhost:3090/api/chargebacks/2",
        "description": "Default"
      },
      "resource": {
        "href": "http://localhost:3090/api/data_stores/12",
        "name": "EXPORT"
      }
    }
  ]
}
```

```
]
}
```

Basic Template for single chargeback request

POST /api/chargebacks/<chargeback_id>

```
{
  "action" : "assign",
  "assignments" : [
    {
      <resource>
    }
  ]
}
```

Basic Template for chargebacks collection request

POST /api/chargebacks

```
{
  "action" : "assign",
  "assignments" : [
    {
      "chargeback":
      {
        <chargeback>
      },
      <resource>
    }
  ]
}
```

<chargeback> is same for all types of assignments. For <chargeback> can be used href or id:

- {"href": "http://localhost:3090/api/chargebacks/1"}
- {"id": "1"}

<resource> can be different across types of assignments - formats are described in next section.

Enterprise

<resource>:

"href": "http://localhost:3090/api/enterprises/1"

POST /api/chargebacks

```
{
  "action" : "assign",
  "assignments" : [
    {
      "chargeback":
      {
        "id" : 2
      },
      "resource":
      {
        "href": "http://localhost:3090/api/enterprises/1"
      }
    }
  ]
}
```

```
]
}
```

Response

```
{
  "success": true,
  "message": "Rates assigned successfully",
  "result": [
    {
      "chargeback": {
        "href": "http://localhost:3090/api/chargebacks/2",
        "description": "Default"
      },
      "resource": {
        "href": "http://localhost:3090/api/enterprises/1",
        "name": "Enterprise"
      }
    }
  ]
}
```

Selected Providers

<resource>:

```
"href": "http://localhost:3090/api/enterprises/1"
```

Selected Clusters

<resource>:

```
"href": "http://localhost:3090/api/clusters/1"
```

Tagged VMs and Instances

Identification can be done by tag's name and tag's category, by `id` or by `href`.

`assignment_prefix` is string which partly determines type of assignment. In this case for "VMs and Instances" value for `assignment_prefix` is `vm`.

<resource>:

```
- "tag": {
  "category" : "department",
  "name" : "finance",
  "assignment_prefix": "vm"
}
- "tag": {
  "id" : 28,
  "assignment_prefix": "vm"
}
- "tag": {
  "href" : 'http://localhost:3090/api/tags/28',
  "assignment_prefix": "vm"
}
```

Request

POST /api/chargebacks

```
{
  "action" : "assign",
```



```

"assignments" : [
  {
    "chargeback": { "id" : 1},
    "tag": { "category" : "department", "name" : "finance",
"assignment_prefix": "vm" }
  },
  {
    "chargeback": { "id" : 1},
    "tag": { "id" : 2}
  },
  {
    "chargeback": { "id" : 3},
    "tag": { "href" : "http://localhost:3090/api/tags/28", "assignment_prefix":
"vm" }
  }
]
}

```

Tagged Container Images

Identification can be done by tag's name and tag's category, by **id** or by **href**.

assignment_prefix is string which partly determines type of assignment. - In this case for "Container Images" value for **assignment_prefix** is **container_image**

<resource>:

```

"tag":
{
  "category" : "department",
  "name" : "finance",
  "assignment_prefix": "container_image"
}

```

or

```

"tag":
{
  "id" : 28,
  "assignment_prefix": "container_image"
}

```

or

```

"tag":
{
  "href" : 'http://localhost:3090/api/tags/28',
  "assignment_prefix": "container_image"
}

```

Request

POST /api/chargebacks

```

{
  "action" : "assign",
  "assignments" : [
    {
      "chargeback": { "id" : 1},
      "tag": { "category" : "department", "name" : "finance",
"assignment_prefix": "vm" }
    },
    {

```

```

    "chargeback": { "id" : 1},
    "tag": { "id" : 2}
  },
  {
    "chargeback": { "id" : 3},
    "tag": { "href" : "http://localhost:3090/api/tags/28", "assignment_prefix":
"vm" }
  }
]
}

```

Labeled Container Images

<resource>:

"href": "http://localhost:3090/api/container_images/934/custom_attributes/30331"

This resource represents custom attributes on container images, so called “labels”.

Request

POST /api/chargebacks

```

{
  "action" : "assign",
  "assignments" :
  [
    {
      "chargeback": { "id" : 1},
      "resource": { "href":
"http://localhost:3090/api/container_images/934/custom_attributes/30331" }
    },
    {
      "chargeback": { "id" : 1},
      "resource": { "href":
"http://localhost:3090/api/container_images/934/custom_attributes/30324" }
    }
  ]
}

```

Tenants

<resource>:

"href": "http://localhost:3090/api/tenants/1"

Example

Request

POST /api/chargebacks

```

{
  "action" : "assign",
  "assignments" : [
    {
      "chargeback": {
        "id" : 2
      },
      "resource": {
        "href": "http://localhost:3090/api/tenants/1"
      }
    },
    {

```

```

        "chargeback": {
            "id" : 2
        },
        "resource": {
            "href": "http://localhost:3090/api/tenants/2"
        }
    }
]
}

```

Response

```

{
    "success": true,
    "message": "Rates assigned successfully",
    "result": [
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/1",
                "name": "My Company"
            }
        },
        {
            "chargeback": {
                "href": "http://localhost:3090/api/chargebacks/2",
                "description": "Default"
            },
            "resource": {
                "href": "http://localhost:3090/api/tenants/2",
                "name": "OpenStack Cloud Provider OpenStack"
            }
        }
    ]
}

```

Enterprise

<resource>:

```

"resource" :
{
    href": "http://localhost:3090/api/enterprises/1"
}

```

Request

POST /api/chargebacks

```

{
    "action" : "assign",
    "assignments" : [
        {
            "chargeback":
            {
                "id" : 2
            },
            "resource":
            {
                "href": "http://localhost:3090/api/enterprises/1"
            }
        }
    ]
}

```

```
    }  
  ]  
}
```

Selected Datastores

<resource>:

```
"resource":  
{  
  "href": "http://localhost:3090/api/enterprises/1"  
}
```

Example

Request

POST /api/chargebacks

```
{  
  "action" : "assign",  
  "assignments" : [  
    {  
      "chargeback" :  
        {  
          "id" : 2  
        },  
      "resource" :  
        {  
          "href" : "/api/data_stores/1"  
        }  
    }  
  ]  
}
```

Response

```
{  
  "success": true,  
  "message": "Rates assigned successfully",  
  "result": [  
    {  
      "chargeback": {  
        "href": "http://localhost:3090/api/chargebacks/2",  
        "description": "Default"  
      },  
      "resource": {  
        "href": "http://localhost:3090/api/data_stores/1",  
        "name": "Shared-NFS"  
      }  
    }  
  ]  
}
```

Tagged Datastores

<resource>:

```
"tag":  
{  
  "href": "http://localhost:3090/api/tags/26",  
  "assignment_prefix": "storage"  
}
```

or

```
"tag":
{
  "category" : "department",
  "name" : "finance",
  "assignment_prefix": "storage"
}
```

or

```
"tag":
{
  "id" : "26",
  "assignment_prefix": "storage"
}
```

Example

Request

POST /api/chargebacks

```
{
  "action" : "assign",
  "assignments" : [
    {
      "tag": {
        "category" : "department",
        "name" : "finance",
        "assignment_prefix": "storage"
      }
    }
  ]
}
```

Response

```
{
  "success": true,
  "message": "Rates assigned successfully",
  "result": [
    {
      "chargeback": {
        "href": "http://localhost:3090/api/chargebacks/2",
        "description": "Default"
      },
      "tag": {
        "href": "http://localhost:3090/api/tags/48",
        "name": "finance",
        "category": "department",
        "assignment_prefix": "storage"
      }
    }
  ]
}
```

Tenants

<resource>:

"href": "http://localhost:3090/api/tenants/1"

Request

POST /api/chargebacks

```
{
  "action": "assign",
  "assignments": [
    {
      "chargeback": {
        "id": 2
      },
      "resource": {
        "href": "http://localhost:3090/api/tenants/1"
      }
    },
    {
      "chargeback": {
        "id": 2
      },
      "resource": {
        "href": "http://localhost:3090/api/tenants/2"
      }
    }
  ]
}
```

Response

```
{
  "success": true,
  "message": "Rates assigned successfully",
  "result": [
    {
      "chargeback": {
        "href": "http://localhost:3090/api/chargebacks/2",
        "description": "Default"
      },
      "resource": {
        "href": "http://localhost:3090/api/tenants/1",
        "name": "My Company"
      }
    },
    {
      "chargeback": {
        "href": "http://localhost:3090/api/chargebacks/2",
        "description": "Default"
      },
      "resource": {
        "href": "http://localhost:3090/api/tenants/2",
        "name": "OpenStack Cloud Provider OpenStack"
      }
    }
  ]
}
```

Unassign action for Chargeback Rates

Unassign action has same signature as assign action and action is 'unassign'.

Example

Request

POST

```
{
  "action" : "unassign",
  "assignments" : [
    {
      "chargeback": { "id" : 3},
      "tag": { "category" : "prov_max_retirement_days", "name" : "30",
"assignment_prefix": "vm" }
    }
  ]
}
```

Example of Chargeback Flow

- [Create Chargeback Report](#)
- [Create Chargeback Rates](#)
- [Assign Chargeback Rates](#)
- [Run Chargeback Report](#)
- [Results of Chargeback Report](#)

Create Chargeback Report

POST api/reports

```
{
  "action" : "import",
  "resource" : {
    "report" : {
      "rpt_group" : "Custom",
      "rpt_type" : "Custom",
      "menu_name" : "Chargeback Flow Example",
      "col_order" : ["vm_name", "display_range", ..],
      "cols" : ["start_date", "display_range", "vm_name", ..],
      "headers" : ["VM Name", "Date Range", ..],
      "title" : "Chargeback Flow Example",
      "db" : "ChargebackVm",
      "group" : "y",
      "sortby" : ["vm_name", "start_date"],
      "template_type" : "report",
      "order" : "Ascending",
      "db_options" : {
        "rpt_type" : "ChargebackVm",
        "options" : {
          <report_options>
        }
      }
    },
    "options" : { "save" : true }
  }
}
```

Attribute **"report"** contains JSON with report definition which can be obtained by exporting a chargeback report.

- **db** is type of report for chargeback purposes. Possible values are **ChargebackVm**, **ChargebackContainerProject**, **ChargebackContainerImage**.

Chargeback Report Options

Chargeback Report Options are present on index **options** in request json. There are also some options about filtering and grouping which are described in this section. Default attributes are described first and then there is explanation of attributes according to chargeback type of report(attribute **db**).

Example Request with **options**

POST /api/chargebacks

```
"db_options" : {
  "rpt_type" : "ChargebackVm",
  "options" : {
    "interval" : "monthly",
    "interval_size" : 12,
    "end_interval_offset" : 1,
    "tenant_id" : "1",
    "method_for_allocated_metrics" : "max",
    "include_metrics" : true,
    "cumulative_rate_calculation" : false,
    "groupby" : "date",
    "groupby_tag" : null,
    "groupby_label" : null
  }
},
...
```

- **method_for_allocated_metrics** - max or avg
- **include_metrics** - true or false - enable/disable whether to include metrics data, when disable it uses current values of metric data or charged resources
- **cumulative_rate_calculation** - true or false enable/disable cumulative calculation for tagged resources
- **interval** is which time detail is used for the report. Possible values are **monthly**, **weekly**, **daily**

Metric data for chargeback report are taken from this interval:

```
`<Time.now-interval_size, Time.now-end_interval_offset>`
```

- **interval_size**, **end_interval_offset** - attributes in **options**. there is used unit according to interval parameter, for example: when interval is “daily” then **interval_size=3** and **end_interval_offset=2** mean that generated report will be based on metrics starting 3 days ago to 1 day ago.

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      ...
    }
  },
  ...
}
```

Filtering according to resource

This is part of json used in request below.

Each section lists possible options to filter according to resource's type. There are used **ids** for identification each type of resource and those **ids** are used in attributes according to type of resources. Name of attributes can be find in following sections. Attributes are located in json for request(example with tenant) on indexes **report**,

db_options and **options**:

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      "tenant_id" : 1,
      ...
    }
  },
  ...
}
```

Chargeback for VMs (db=ChargebackVm)

tenant_id - Tenant id

or

owner - **userid** of user which owns VMs.

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      "owner" : "admin",
      ...
    }
  },
  ...
}
```

or

tag - tag_path of tag

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      "tag" : "/managed/prov_max_cpu/2",
      ...
    }
  },
  ...
}
```

Chargeback for Projects (db=ChargebackContainerProject)

tag - tag_path of tag

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      "tag" : "/managed/prov_max_cpu/2",
      ...
    }
  },
  ...
}
```

or

provider_id - "all" or Openshift Provider id

entity_id - "all" or Container Project id

```
"report" : {
  "db_options" : {
    "options" : {
      "provider_id" : "41",
      "entity_id" : "21",
      ...
    }
  },
  ...
}
```

Chargeback for Container Images (db=ChargebackContainerImage)

provider_id - "all" or Openshift Provider id

entity_id - "all" or Container Image id

Request

POST /api/chargebacks

```
"report" : {
  "db_options" : {
    "options" : {
      "provider_id" : "all",
      "entity_id" : "all",
      ...
    }
  }
}
```

Grouping Report Options

Chargeback for VMs (db=ChargebackVm)

groupby - specifies a grouping option

Possible values for **groupby**:

date - grouping by date and Vms

date-only - grouping only by date

tenant - grouping by tenants

tag - name of tag category - attribute **groupBy_tag** needs to be also specified. Values for **groupBy_tag** are names of tag category which can be retrieved in **resources**

GET

/api/categories?expand=resources&attributes=name

Chargeback for Projects (db=ChargebackContainerProject)

groupBy - specifies a grouping option

Possible values for **groupBy**:

date - grouping by date and container projects

date-only - grouping only by date

project - grouping by container projects

tag - name of tag category - attribute **groupBy_tag** needs to be also specified. Values for **groupBy_tag** are names of tag category which can be retrieved in **resources**

GET

/api/categories?expand=resources&attributes=name

Chargeback for Container Images (db=ChargebackContainerImage)

groupBy - specifies a grouping option

Possible values for **groupBy**:

date - grouping by date and containers

date-only - grouping only by date

project - grouping by container images

label - grouping by container labels and when this value is set - attribute **groupBy_tag** needs to be also specified. Values for **groupBy_tag** are names of labels which can be retrieved by **api/container_images/:id/custom_attributes** endpoint (attribute=name) for single container image.

Example request

GET

**/api/container_images/934/custom_attributes?
expand=resources&attributes=name&filter[]=section=docker_labels**

Response

```
{  
  "name": "custom_attributes",  
  "count": 2041860,  
}
```

```

    "subcount": 11,
    "subquery_count": 11,
    "pages": 1,
    "resources": [
      {
        "href":
"http://localhost:3090/api/container_images/934/custom_attributes/30322",
        "name": "build-date",
        "id": "30322"
      },
      {
        "href":
"http://localhost:3090/api/container_images/934/custom_attributes/30324",
        "name": "architecture",
        "id": "30324"
      }
    ],
    ...
  }
}

```

Example of request to create chargeback report

Request

POST /api/chargebacks

```

{
  "action" : "import",
  "resource" : {
    "report" : {
      "rpt_group" : "Custom",
      "rpt_type" : "Custom",
      "menu_name" : "Chargeback Flow Example 1",
      "col_order" : ["vm_name", "display_range", "chargeback_rates",
"memory_allocated_metric", "memory_allocated_cost", "memory_allocated_rate",
"memory_cost"],
      "cols" : ["start_date", "display_range", "vm_name",
"chargeback_rates", "memory_allocated_metric", "memory_allocated_cost",
"memory_allocated_rate", "memory_cost"],
      "headers" : ["VM Name", "Date Range", "Chargeback Rates", "Memory
Allocated over Time Period", "Memory Allocated Cost", "Memory Allocated
Rate", "Memory Total Cost"],
      "title" : "Chargeback Flow Example 1",
      "db" : "ChargebackVm",
      "group" : "y",
      "sortby" : ["vm_name", "start_date"],
      "template_type" : "report",
      "order" : "Ascending",
      "db_options" : {
        "rpt_type" : "ChargebackVm",
        "options" : {
          "interval" : "monthly",
          "interval_size" : 12,
          "end_interval_offset" : 1,
          "tenant_id" : "1",
          "method_for_allocated_metrics" : "max",
          "include_metrics:" : true,
          "cumulative_rate_calculation" : false,
          "groupby" : "date",
          "groupby_tag" : null,
          "groupby_label" : null
        }
      }
    }
  }
}

```

```

    },
    "options" : { "save" : true }
  }
}

```

Response

```

{
  "results": [
    {
      "success": true,
      "message": "Imported Report: [Chargeback Flow Example 1]",
      "result": {
        "id": "553",
        "name": "Chargeback Flow Example 1",
        "title": "Chargeback Flow Example 1",
        "rpt_group": "Custom",
        "rpt_type": "Custom",
        "priority": null,
        "db": "ChargebackVm",
        "cols": [
          "start_date",
          "display_range",
          "vm_name",
          "chargeback_rates",
          "memory_allocated_metric",
          "memory_allocated_cost",
          "memory_allocated_rate",
          "memory_cost"
        ],
        "include": null,
        "col_order": [
          "vm_name",
          "display_range",
          "chargeback_rates",
          "memory_allocated_metric",
          "memory_allocated_cost",
          "memory_allocated_rate",
          "memory_cost"
        ],
        "headers": [
          "VM Name",
          "Date Range",
          "Chargeback Rates",
          "Memory Allocated over Time Period",
          "Memory Allocated Cost",
          "Memory Allocated Rate",
          "Memory Total Cost"
        ],
        "conditions": null,
        "order": "Ascending",
        "sortby": [
          "vm_name",
          "start_date"
        ],
        "group": "y",
        "graph": null,
        "dims": null,
        "created_on": "2020-05-19T12:16:31Z",
        "updated_on": "2020-05-19T12:16:31Z",
        "filename": null,
        "file_mtime": null,
        "categories": null,
        "timeline": null,

```

```

        "template_type": "report",
        "where_clause": null,
        "db_options": {
            "rpt_type": "ChargebackVm",
            "options": {
                "interval": "monthly",
                "interval_size": 12,
                "end_interval_offset": 1,
                "tenant_id": "1",
                "method_for_allocated_metrics": "max",
                "include_metrics": true,
                "cumulative_rate_calculation": false,
                "groupby": "date",
                "groupby_tag": null,
                "groupby_label": null
            }
        },
        "generate_cols": null,
        "generate_rows": null,
        "col_formats": null,
        "tz": null,
        "time_profile_id": null,
        "display_filter": null,
        "col_options": null,
        "rpt_options": null,
        "miq_group_id": "2",
        "user_id": "1"
    }
}
]
}

```

Create Chargeback Rates

Chargeback rate is complex structure. Create chargeback rate requires two endpoints: `/api/chargebacks` and `/api/rates`. Endpoint `/api/chargebacks` manage top level rate entity which aggregate other relations like rate details(`api/rates`), currencies(`api/currencies`), measures(`api/measures`), chargeable_fields(`api/chargeable_fields`) and chargeback tiers. This entity is also used to [assigning rates](#) to any resource.

Create Chargebacks (top-level rate entity)

Request

POST `/api/chargebacks`

```

{
  "guid" : "daily",
  "description" : "1",
  "rate_type" : "Compute"
}

```

Response

```

{
  "results": [
    {
      "href": "http://localhost:3090/api/chargebacks/14",
      "id": "14",
      "guid": "daily",
      "description": "1",
      "rate_type": "Compute",
    }
  ]
}

```

```

        "created_on": "2020-05-26T12:15:51Z",
        "updated_on": "2020-05-26T12:15:51Z",
        "default": false
    }
]
}

```

Now we need to create rate details(**api/rates**) which includes chargeable_fields, currencies, measures and tiers. It is also important assign rate details to chargeback top level entity by **chargeback_rate** attribute.

Request

POST /api/rates

```

{
  "per_time" : "daily",
  "chargeback_rate" : {"href": "http://localhost:3090/api/chargebacks/14"},
  "chargeable_field_id" : "1",
  "description": "My CPU allocation rate",
  "group" : "cpu",
  "per_unit" : "megahertz",
  "source" : "allocated",
  "chargeback_rate_detail_measure_id": 1,
  "detail_currency" : {"id": 5},
  "detail_measure" : {"href": "http://localhost:3090/api/measures/3"},
  "chargeable_field" : {"href": "http://localhost:3090/api/chargeable_fields/3"},
  "chargeback_tiers" : [
    {
      "start": 0.0,
      "finish": 1,
      "fixed_rate": 2.0,
      "variable_rate": 0.0
    },
    {
      "start": 1,
      "finish": "infinity",
      "fixed_rate": 2.0,
      "variable_rate": 0.0
    }
  ]
}

```

Response

```

{
  "results": [
    {
      "href": "http://localhost:3090/api/rates/1862",
      "id": "1862",
      "enabled": true,
      "description": "My CPU allocation rate",
      "group": "cpu",
      "source": "allocated",
      "metric": null,
      "per_time": "daily",
      "per_unit": "megahertz",
      "friendly_rate": "Daily @ 2.0 + 0.0 per Megahertz from 0.0 to 1.0\nDaily @ 2.0 + 0.0 per Megahertz from 1.0 to Infinity",
      "chargeback_rate_id": "14",
      "created_on": "2020-05-26T12:36:32Z",
      "updated_on": "2020-05-26T12:36:32Z",
      "chargeback_rate_detail_measure_id": "3",
      "chargeback_rate_detail_currency_id": "5",

```

```

        "chargeable_field_id": "3",
        "sub_metric": null
    }
]
}

```

Assign Chargeback Rate

There are examples in those sections:

- [Querying Assigned Chargeback Rates](#)
- [Assign action for Chargeback Rates](#)
- [Unassign action for Chargeback Rates](#)

Run Chargeback Report

Please refer to [Run Report](#) section for details on how to run chargeback report. Running report requires report id or report href:

Report id or report href can be obtained by [chargeback report creation](#) or by listing chargeback reports:

Request

```

GET /api/reports?
expand=resources&attributes=db&filter[]=db='ChargebackVm'&filter[]=or
db='ChargebackContainer%'

```

Response

```

{
  "name": "reports",
  "count": 519,
  "subcount": 47,
  "subquery_count": 47,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3090/api/reports/157",
      "db": "ChargebackContainerImage",
      "id": "157"
    },
    {
      "href": "http://localhost:3090/api/reports/152",
      "db": "ChargebackVm",
      "id": "152"
    },
    ...
  ]
}

```

Results of Chargeback Report

There is request which list of report result.

Allowed parameters:

hash_attribute=value - enables hash sorting and pagination for value column

sort_by - column from **MiqReportResult#report#col_order**

sort_order - asc or desc

limit - count of records for one page

offset - result will start from <offset>. record

filter_column - chosen column for filtering

filter_string - this string value is used for filtering on <filter_column>, record of report result will be returned in API when <filter_string> is contained in value of <filter_column>.

There is possibility to filter according to more columns. We need to add other filter parameters with suffix **_x** where **x** is number

(...&filter_column_1=name&filter_string_1=TIE&filter_column_2=hostname&filter_string_2=test)

Request

GET /api/results/23039?

hash_attribute=result_set&sort_by=&sort_order=&limit=20&offset=0&filter_string=windemo

Response

```
{
  "href": "http://localhost:3090/api/results/23039",
  "id": "23039",
  "name": "Chargeback - Demo",
  "...",
  "result_set": [
    {
      "start_date": "01/01/19 00:00:00 UTC",
      "display_range": "Jan 2019",
      "vm_name": "windemo",
      "owner_name": "Cloud Operators",
      "provider_name": "vCenter",
      "total_cost": "$48,303.51",
      "cpu_cost": "$935.65",
      "cpu_used_cost": "$191.65",
      "cpu_used_metric": "12.88 MHz",
      "memory_cost": "$1,609.69",
      "memory_used_cost": "$1,609.69",
      "memory_used_metric": "108.18 MB",
      "net_io_used_cost": "$372.00",
      "net_io_used_metric": "0.00 KBps",
      "chargeback_rates": "Default"
    },
    {
      "start_date": "02/01/19 00:00:00 UTC",
      "display_range": "Feb 2019",
      "vm_name": "windemo",
      "owner_name": "Cloud Operators",
      "provider_name": "vCenter",
      "total_cost": "$43,851.09",
      "cpu_cost": "$861.44",
      "cpu_used_cost": "$189.44",
      "cpu_used_metric": "14.09 MHz",
      "memory_cost": "$1,659.73",
      "memory_used_cost": "$1,659.73",
      "memory_used_metric": "123.49 MB",
      "net_io_used_cost": "$336.00",
      "net_io_used_metric": "0.00 KBps",
      "chargeback_rates": "Default"
    }
  ],
}
```

```

    {
      "start_date": "03/01/19 00:00:00 UTC",
      "display_range": "Mar 2019",
      "vm_name": "windemo",
      "owner_name": "Cloud Operators",
      "provider_name": "vCenter",
      "total_cost": "$28,906.99",
      "cpu_cost": "$553.43",
      "cpu_used_cost": "$113.43",
      "cpu_used_metric": "12.89 MHz",
      "memory_cost": "$1,231.94",
      "memory_used_cost": "$1,231.94",
      "memory_used_metric": "139.99 MB",
      "net_io_used_cost": "$220.00",
      "net_io_used_metric": "0.00 KBps",
      "chargeback_rates": "Default"
    }
  ],
  "count": 3,
  "subcount": 3,
  "pages": 1
}

```

- [Cloud Volumes](#)
 - [Querying Cloud Volumes](#)
 - [Deleting Cloud Volumes](#)

Cloud Volumes

Management of Cloud Volumes is provided via the following collection:

`/api/cloud_volumes`

The following actions are available on cloud_volumes:

- [Querying Cloud Volumes](#)
- [Deleting Cloud Volumes](#)

Querying Cloud Volumes

Querying all cloud volumes in the system is simply:

`GET /api/cloud_volumes`

And getting details of a specific Cloud Volume is as follows:

`GET /api/cloud_volumes/:id`

Deleting Cloud Volumes

Cloud Volumes can be deleted via either the **delete** POST action or via the DELETE HTTP method.

`POST /api/cloud_volumes/101`

```

{
  "action" : "delete"
}

```

or simply:

```
DELETE /api/cloud_volumes/101
```

Deleting multiple Cloud Volumes can be done as follows:

```
POST /api/cloud_volumes
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/cloud_volumes/101" },
    { "href" : "http://localhost:3000/api/cloud_volumes/102" },
    ...
  ]
}
```

- [Conditions](#)
 - [Querying Conditions](#)
 - [Creating Conditions](#)
 - [Editing Conditions](#)
 - [Deleting Conditions](#)

Conditions

Management of conditions is provided via the following collection:

```
/api/conditions
```

Full CRUD actions on conditions is available:

- [Querying Conditions](#)
- [Creating Conditions](#)
- [Editing Conditions](#)
- [Deleting Conditions](#)

Querying Conditions

Querying all conditions in the system is simply:

```
GET /api/conditions
```

Getting details on a specific Condition:

```
GET /api/conditions/:id
```

Creating Conditions

Conditions can be created via a POST to the conditions collection or via the **create** action signature which also allows creation of multiple conditions in a single request.

```
POST /api/conditions
```

```
{
  "name" : "sample_condition",
  "description" : "Sample Condition",
  "expression" : { "=" : { "field" : "ExtManagementSystem.container_images-
architecture", "value" : "dsa" } },
  "towhat" : "ExtManagementSystem",

```

```
"modifier" : "allow"
}
```

or creating multiple conditions:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "condition_1", "description" : "Condition 1", ... },
    { "name" : "condition_2", "description" : "Condition 2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating conditions.

Editing Conditions

POST /api/conditions/:id

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Condition Description"
  }
}
```

or editing multiple conditions:

POST /api/conditions

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/conditions/101",
      "description" : "Updated Condition Description 1"
    },
    {
      "href" : "http://localhost:3000/api/conditions/102",
      "description" : "Updated Condition Description 2"
    },
    ...
  ]
}
```

Deleting Conditions

Conditions can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/conditions/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/conditions/101

Deleting multiple conditions can be done as follows:

POST /api/conditions

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/conditions/101" },
    { "href" : "http://localhost:3000/api/conditions/102" },
    ...
  ]
}
```

- [Configuration Script Payloads](#)
 - [Querying Configuration Script Payloads](#)
 - [Editing configuration script payloads](#)
 - [Updating configuration script payloads](#)
 - [Patch method:](#)
 - [Querying Configuration Script Payload Authentication](#)
 - [Creating Configuration Script Payload Authentication](#)

Configuration Script Payloads

Management of configuration script payloads is provided through the following collection:

/api/configuration_script_payloads

The following actions on configuration script payloads are available:

- [Querying Configuration Script Payloads](#)
- [Editing Configuration Script Payloads](#)
- [Updating Configuration Script Payloads](#)
- [Querying Configuration Script Payload Authentication](#)
- [Creating Configuration Script Payload Authentication](#)

Querying Configuration Script Payloads

Query all configuration script payloads in the system:

GET /api/configuration_script_payloads

Example response:

```
{
  "name": "configuration_script_payloads",
  "count": 2,
  "subcount": 2,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    },
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    }
  ],
  "actions": [
    {
```

```

    "name": "edit",
    "method": "post",
    "href": "http://localhost:3000/api/configuration_script_payloads"
  }
],
"links": {
  "self": "http://localhost:3000/api/configuration_script_payloads?offset=0",
  "first": "http://localhost:3000/api/configuration_script_payloads?offset=0",
  "last": "http://localhost:3000/api/configuration_script_payloads?offset=0"
}
}

```

Query the details on a specific configuration script payload:

GET /api/configuration_script_payloads/:id

Example response:

```

{
  "href": "http://localhost:3000/api/configuration_script_payloads/1",
  "id": "1",
  "manager_id": "2",
  "manager_ref": null,
  "name": "provision-vm-service/list-providers.asl",
  "description": null,
  "variables": null,
  "created_at": "2023-11-08T17:43:30Z",
  "updated_at": "2023-11-08T17:46:03Z",
  "survey_spec": null,
  "inventory_root_group_id": null,
  "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
  "parent_id": null,
  "configuration_script_source_id": "11",
  "run_by_userid": null,
  "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\":\n  \"ListProviders\",\n  \"States\": {\n    \"ListProviders\": {\n      \"Type\":\n      \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n        \"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\": \"$.api_bearer_token\",\n        \"Parameters\": {\n          \"API_URL.$\": \"$.Execution.manageiq_api_url\",\n          \"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n          \"PROVIDER_TYPE\": \"ManageIQ::Providers::Vmware::InfraManager\"\n        }\n      }\n    }\n  }\n}",
  "payload_type": "json",
  "credentials": null,
  "context": null,
  "output": null,
  "status": null,
  "miq_task_id": null,
  "payload_valid": true,
  "payload_error": null,
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    },
    {
      "name": "edit",
      "method": "patch",
      "href": "http://localhost:3000/api/configuration_script_payloads/1"
    }
  ]
}

```

```

    "name": "edit",
    "method": "put",
    "href": "http://localhost:3000/api/configuration_script_payloads/1"
  }
]
}

```

Editing configuration script payloads

Configuration script payloads can be edited by using a POST to the configuration script payloads collection:

POST /api/configuration_script_payloads

POST /api/configuration_script_payloads

```

{
  "action": "edit",
  "resources": [
    {"id": 1, "description": "A Helpful Description"}
  ]
}

```

Example response:

```

{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
      "created_at": "2023-11-08T17:43:30Z",
      "updated_at": "2023-11-15T20:14:28Z",
      "survey_spec": null,
      "inventory_root_group_id": null,
      "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
      "parent_id": null,
      "configuration_script_source_id": "2",
      "run_by_userid": null,
      "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\":\n  \"ListProviders\", \n  \"States\": {\n    \"ListProviders\": {\n      \"Type\":\n      \"Task\", \n      \"Resource\": \"docker://docker.io/agrare/list-providers:latest\", \n      \"End\": true, \n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\", \n        \"api_password.$\": \"$.api_password\", \n        \"api_token.$\": \"$.api_token\", \n        \"api_bearer_token.$\":\n        \"$.api_bearer_token\", \n        \"Parameters\": {\n          \"API_URL.$\":\n          \"$.Execution_manageiq_api_url\", \n          \"VERIFY_SSL.$\":\n          \"$.dialog_dialog_verify_ssl\", \n          \"PROVIDER_TYPE\":\n          \"ManageIQ::Providers::Vmware::InfraManager\" \n        } \n      } \n    } \n  } \n  }\n",
      "payload_type": "json",
    }
  ]
}

```

```

        "context": null,
        "output": null,
        "status": null,
        "miq_task_id": null,
        "payload_valid": true,
        "payload_error": null
    }
]
}

```

For instance, configuration script payloads can be edited to map credentials:

POST /api/configuration_script_payloads

```

{
  "action": "edit",
  "resources": [
    {
      "id": 1,
      "credentials": [
        {"my-cred-user": {"credential_ref": "my-credential", "credential_field": "userid"}},
        {"my-cred-password": {"credential_ref": "my-credential", "credential_field": "password"}}
      ]
    }
  ]
}

```

Updating configuration script payloads

Configuration script payloads can be updated by using a PUT method or PATCH method.

PUT method:

PUT /api/configuration_script_payloads/:id

Example request:

PUT /api/configuration_script_payloads/1 {"description": "A Helpful Description"}

Response:

```

{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
    }
  ]
}

```



```

    "created_at": "2023-11-08T17:43:30Z",
    "updated_at": "2023-11-15T20:14:28Z",
    "survey_spec": null,
    "inventory_root_group_id": null,
    "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
    "parent_id": null,
    "configuration_script_source_id": "2",
    "run_by_userid": null,
    "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\":\n  \"ListProviders\",\n  \"States\": {\n    \"ListProviders\": {\n      \"Type\":\n      \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n        \"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\": \"$.api_bearer_token\",\n        \"Parameters\": {\n          \"API_URL.$\": \"$.Execution.manageiq_api_url\",\n          \"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n          \"PROVIDER_TYPE\": \"ManageIQ::Providers::Vmware::InfraManager\"\n        }\n      }\n    }\n  }\n}"
  "payload_type": "json",
  "context": null,
  "output": null,
  "status": null,
  "miq_task_id": null,
  "payload_valid": true,
  "payload_error": null
}
]
}

```

Patch method:

PATCH /api/configuration_script_payloads/:id

Example request:

PATCH /api/configuration_script_payloads/1 {"description": "A Helpful Description"}

Response:

```

{
  "results": [
    {
      "href": "http://localhost:3000/api/configuration_script_payloads/1",
      "description": "A Helpful Description",
      "credentials": {
        "api_user.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "userid"
        },
        "api_password.$": {
          "credential_ref": "manageiq_api",
          "credential_field": "password"
        }
      },
      "id": "1",
      "manager_id": "2",
      "manager_ref": null,
      "name": "provision-vm-service/list-providers.asl",
      "variables": null,
      "created_at": "2023-11-08T17:43:30Z",
      "updated_at": "2023-11-15T20:14:28Z",
      "survey_spec": null,
    }
  ]
}

```

```

    "inventory_root_group_id": null,
    "type": "ManageIQ::Providers::Workflows::AutomationManager::Workflow",
    "parent_id": null,
    "configuration_script_source_id": "2",
    "run_by_userid": null,
    "payload": "{\n  \"Comment\": \"List Providers.\",\n  \"StartAt\":\n  \"ListProviders\",\n  \"States\": {\n    \"ListProviders\": {\n      \"Type\":\n      \"Task\",\n      \"Resource\": \"docker://docker.io/agrare/list-providers:latest\",\n      \"End\": true,\n      \"Credentials\": {\n        \"api_user.$\": \"$.api_user\",\n        \"api_password.$\": \"$.api_password\",\n        \"api_token.$\": \"$.api_token\",\n        \"api_bearer_token.$\": \"$.api_bearer_token\",\n        \"Parameters\": {\n          \"API_URL.$\": \"$.Execution._manageiq_api_url\",\n          \"VERIFY_SSL.$\": \"$.dialog.dialog_verify_ssl\",\n          \"PROVIDER_TYPE\": \"ManageIQ::Providers::Vmware::InfraManager\"\n        }\n      }\n    }\n  }\n}",
    "payload_type": "json",
    "context": null,
    "output": null,
    "status": null,
    "miq_task_id": null,
    "payload_valid": true,
    "payload_error": null
  }
]
}

```

Querying Configuration Script Payload Authentication

Query the authentication credentials for a specific configuration script payload:

GET /api/configuration_script_payloads/:id/authentications

Example response:

```

{
  "name": "authentications",
  "count": 7,
  "subcount": 1,
  "pages": 1,
  "resources": [
    {
      "href":
"http://localhost:3000/api/configuration_script_payloads/2/authentications/3"
    }
  ],
  "actions": [
    {
      "name": "create",
      "method": "post",
      "href":
"http://localhost:3000/api/configuration_script_payloads/2/authentications"
    }
  ],
  "links": {
    "self":
"http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0",
    "first":
"http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0",
    "last":
"http://localhost:3000/api/configuration_script_payloads/2/authentications?offset=0"
  }
}

```

```
}  
}
```

Get the details on a specific authentication credential for a configuration script payload:

```
GET /api/configuration_script_payloads/:id/authentications/:id
```

Example response:

```
{  
  "href":  
    "http://localhost:3000/api/configuration_script_payloads/2/authentications/3",  
  "id": "3",  
  "name": "manageiq_api",  
  "authtype": null,  
  "userid": "admin",  
  "password": null,  
  "resource_id": "25",  
  "resource_type": "ExtManagementSystem",  
  "created_on": "2023-11-15T19:59:23Z",  
  "updated_on": "2023-11-15T19:59:23Z",  
  "last_valid_on": null,  
  "last_invalid_on": null,  
  "credentials_changed_on": null,  
  "status": null,  
  "status_details": null,  
  "type": "ManageIQ::Providers::Workflows::AutomationManager::WorkflowCredential",  
  "auth_key": null,  
  "fingerprint": null,  
  "service_account": null,  
  "public_key": null,  
  "manager_ref": "3",  
  "options": null,  
  "evm_owner_id": null,  
  "miq_group_id": "1",  
  "tenant_id": "1",  
  "become_username": null,  
  "become_password": null,  
  "auth_key_password": null,  
  "ems_ref": "manageiq_api",  
  "become_method": null  
}
```

Creating Configuration Script Payload Authentication

Authentication credentials for a configuration script payload can be created by using a POST to the configuration script payload collection:

```
POST /api/configuration_script_payloads/:id/authentications
```

```
{  
  "action": "create",  
  "description": "Description",  
  "name": "my credential",  
  "type": "ManageIQ::Providers::AnsibleTower::AutomationManager::Credential"  
}
```

- [Configuration Script Sources](#)
 - [Querying Configuration Script Sources](#)
 - [Creating Configuration Script Sources](#)
 - [Editing Configuration Script Sources](#)
 - [Refreshing Configuration Script Sources](#)

- [Request:](#)
- [Response:](#)
- [Deleting Configuration Script Sources](#)

Configuration Script Sources

Management of configuration script sources is provided via the following collection:

`/api/configuration_script_sources`

Full CRUD actions on configuration script sources is available:

- [Querying Configuration Script Sources](#)
- [Creating Configuration Script Sources](#)
- [Editing Configuration Script Sources](#)
- [Refreshing Configuration Script Sources](#)
- [Deleting Configuration Script Sources](#)

Querying Configuration Script Sources

Querying all configuration script sources in the system is simply:

`GET /api/configuration_script_sources`

Getting details on a specific configuration script source:

`GET /api/configuration_script_sources/:id`

Querying configuration script payloads of a particular configuration script source:

`GET /api/configuration_script_sources/:id/configuration_script_payloads?expand=resources`

Querying the configuration script payloads of a particular configuration script source that belong to a specific region:

`GET /api/configuration_script_sources/:id/configuration_script_payloads?expand=resources&filter[]=region_number=10`

Creating Configuration Script Sources

Configuration script sources can be created via a POST to the configuration script sources collection or via the **create** action signature which also allows creation of multiple configuration script sources in a single request.

`POST /api/configuration_script_sources`

```
{
  "name" : "My Project",
  "description" : "My Project's Description",
  "manager_resource" : { "href" : "http://localhost:3000/api/providers/7" },
  "related" : {}
}
```

or creating multiple configuration script sources:

```
{
  "action" : "create",
  "resources" : [
    { "name" : "My First Project", "description" : "My First Project Description",
    ... },
    { "name" : "My Second Project", "description" : "My Second Project
Description", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating configuration script sources.

Editing Configuration Script Sources

POST /api/configuration_script_sources/:id

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Project Description"
  }
}
```

or editing multiple configuration script sources:

POST /api/configuration_script_sources

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/configuration_script_sources/101",
      "description" : "Updated Project Description 1"
    },
    {
      "href" : "http://localhost:3000/api/configuration_script_sources/102",
      "description" : "Updated Project Description 2"
    },
    ...
  ]
}
```

Refreshing Configuration Script Sources

Configuration script sources can be refreshed by posting a **refresh** action to a single resource or to multiple resources in bulk by targeting the collection.

POST /api/configuration_script_sources/:id

```
{
  "action" : "refresh"
}
```

or refreshing multiple configuration script sources:

POST /api/configuration_script_sources

```
{
  "action" : "refresh",
}
```

```

    "resources" : [
      { "id" : "101" },
      { "id" : "102" }
    ]
  }

```

Example:

Request:

POST /api/configuration_script_sources/101

```

{
  "action" : "refresh"
}

```

Response:

```

{
  "success" : true,
  "message" : "Refreshing ConfigurationScriptSource id:101 name:'Test Configuration Script Source'",
  "task_id" : "7",
  "task_href" : "http://localhost:3000/api/tasks/7",
  "tasks" : [
    {
      "id" : "7",
      "href" : "http://localhost:3000/api/tasks/7"
    }
  ]
}

```

Deleting Configuration Script Sources

Configuration script sources can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/configuration_script_sources/101

```

{
  "action" : "delete"
}

```

or simply:

DELETE /api/configuration_script_sources/101

Deleting multiple configuration script sources can be done as follows:

POST /api/configuration_script_sources

```

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/configuration_script_sources/101" },
    { "href" : "http://localhost:3000/api/configuration_script_sources/102" },
    ...
  ]
}

```

- [Custom Actions](#)
 - [Request:](#)
 - [Response:](#)

- [Request:](#)
- [Response:](#)

Custom Actions

With button groups and custom buttons defined on records, the REST API provides visibility to custom actions that represent the custom buttons for the related resources. The custom actions are not returned when querying collections, they are only available when fetching an individual resource, e.g. `/api/:collection/:id`

This is made available with the *custom_actions* and *custom_action_buttons* virtual attributes on resources for the following collections:

/api/cloud_tenants
/api/cloud_volumes
/api/clusters
/api/container_nodes
/api/data_stores
/api/generic_objects
/api/hosts
/api/providers
/api/templates
/api/vms

- *custom_actions* return the custom buttons as laid out in the UI on the resource. The information returned includes any Resource Actions and related Dialogs for the custom button.
- *custom_action_buttons* return the array of custom buttons without the custom button groups hierarchy or details on related dialogs.

Fetching custom actions and custom action buttons for a resource:

GET `/api/:collection/:id?attributes=custom_actions,custom_action_buttons`

Example VM with custom buttons:

Request:

GET `/api/vms/173?attributes=custom_actions,custom_action_buttons`

Response:

```
{
  "href": "https://172.16.30.129/api/vms/173",
  "id": "173",
  "vendor": "vmware",
  "name": "aab-ldap",
  "location": "aab-ldap/aab-ldap.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T20:23:37Z",
  "updated_on": "2018-01-09T20:23:37Z",
  "storage_id": "8",
  "guid": "dee6d969-7c69-44b6-bcd3-3b14f21702c5",
  "ems_id": "1",
  "uid_ems": "420c8f0d-4a8f-0b6d-fc3a-0c3cfab7b33f",
```

```

"tools_status": "toolsNotRunning",
"standby_action": "checkpoint",
"power_state": "off",
...
"custom_actions": {
  "buttons": [
    ],
    "button_groups": [
      {
        "id": "23",
        "name": "VM Actions|Vm|",
        "description": "VM Custom Actions",
        "created_on": "2018-01-09T20:36:50Z",
        "updated_on": "2018-01-09T20:41:40Z",
        "guid": "559b37d1-3232-4c51-8668-f994a5b9149e",
        "set_data": {
          "button_order": [
            1,
            2
          ],
          "button_icon": "pficon pficon-virtual-machine",
          "display": true,
          "applies_to_class": "Vm",
          "group_index": 1
        },
        "buttons": [
          {
            "id": "1",
            "guid": "22b5bf4f-d5d6-43af-bc7b-77460d120065",
            "description": "Analyze Virtual Machine",
            "applies_to_class": "Vm",
            "options": {
              "button_icon": "ff ff-timeline",
              "button_type": "default",
              "display": true,
              "open_url": false,
              "display_for": "single",
              "submit_how": "one"
            },
            "userid": "admin",
            "created_on": "2018-01-09T20:40:11Z",
            "updated_on": "2018-01-09T20:40:11Z",
            "name": "Analyze",
            "visibility": {
              "roles": [
                "_ALL_"
              ]
            },
            "resource_action": {
              "id": "21",
              "resource_id": "1",
              "resource_type": "CustomButton",
              "created_at": "2018-01-09T20:40:11Z",
              "updated_at": "2018-01-09T20:40:11Z",
              "ae_namespace": "SYSTEM",
              "ae_class": "PROCESS",
              "ae_instance": "Automation",
              "ae_attributes": {
                "request": "general/process/analyze"
              }
            },
            "enabled": true,
            "href": "https://172.16.30.129/api/custom_buttons/1"
          }
        ]
      }
    ]
  }
}

```



```

    },
    {
      "id": "2",
      "guid": "e9e77bb3-f52a-4d08-8d58-13923031c6de",
      "description": "Ping Virtual Machine",
      "applies_to_class": "Vm",
      "options": {
        "button_icon": "ff ff-scsi",
        "button_color": "#b4c10f",
        "button_type": "default",
        "display": true,
        "open_url": false,
        "display_for": "single",
        "submit_how": "one"
      },
      "userid": "admin",
      "created_on": "2018-01-09T20:41:40Z",
      "updated_on": "2018-01-09T20:41:40Z",
      "name": "Ping",
      "visibility": {
        "roles": [
          "_ALL_"
        ]
      },
      "resource_action": {
        "id": "22",
        "resource_id": "2",
        "resource_type": "CustomButton",
        "created_at": "2018-01-09T20:41:40Z",
        "updated_at": "2018-01-09T20:41:40Z",
        "ae_namespace": "SYSTEM",
        "ae_class": "PROCESS",
        "ae_instance": "Automation",
        "ae_attributes": {
          "request": "general/virtual_machines/ping"
        }
      },
      "enabled": true,
      "href": "https://172.16.30.129/api/custom_buttons/2"
    }
  ],
  "href": "https://172.16.30.129/api/custom_button_sets/23"
}
]
},
"custom_action_buttons": [
  {
    "href": "https://172.16.30.129/api/custom_buttons/1",
    "id": "1",
    "guid": "22b5bf4f-d5d6-43af-bc7b-77460d120065",
    "description": "Analyze Virtual Machine",
    "applies_to_class": "Vm",
    "options": {
      "button_icon": "ff ff-timeline",
      "button_type": "default",
      "display": true,
      "open_url": false,
      "display_for": "single",
      "submit_how": "one"
    },
    "userid": "admin",
    "created_on": "2018-01-09T20:40:11Z",
    "updated_on": "2018-01-09T20:40:11Z",
    "name": "Analyze",

```

```

    "visibility": {
      "roles": [
        "_ALL_"
      ]
    }
  },
  {
    "href": "https://172.16.30.129/api/custom_buttons/2",
    "id": "2",
    "guid": "e9e77bb3-f52a-4d08-8d58-13923031c6de",
    "description": "Ping Virtual Machine",
    "applies_to_class": "Vm",
    "options": {
      "button_icon": "ff ff-scsi",
      "button_color": "#b4c10f",
      "button_type": "default",
      "display": true,
      "open_url": false,
      "display_for": "single",
      "submit_how": "one"
    },
    "userid": "admin",
    "created_on": "2018-01-09T20:41:40Z",
    "updated_on": "2018-01-09T20:41:40Z",
    "name": "Ping",
    "visibility": {
      "roles": [
        "_ALL_"
      ]
    }
  }
],
"actions": [
  ...
]
}

```

When custom action buttons are defined on a VM, the returned VM resource is extended with custom actions which can be triggered via a POST action.

The action name to trigger the action is the lower-cased name of the custom button.

As an example, here's a VM without custom buttons so all actions returned are the common ones defined by the system:

```

{
  "href": "https://172.16.30.129/api/vms/172",
  "id": "172",
  "vendor": "vmware",
  "name": "aab-idp",
  "location": "aab-idp/aab-idp.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T19:23:37Z",
  "updated_on": "2018-01-09T19:23:37Z",
  "storage_id": "8",
  ...
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {

```

```

    "name": "add_event",
    "method": "post",
    "href": "https://172.16.30.129/api/vms/173"
  },
  ...
  {
    "name": "set_miq_server",
    "method": "post",
    "href": "https://172.16.30.129/api/vms/173"
  },
  {
    "name": "scan",
    "method": "post",
    "href": "https://172.16.30.129/api/vms/173"
  },
  {
    "name": "delete",
    "method": "delete",
    "href": "https://172.16.30.129/api/vms/173"
  }
]
}

```

Here is an example of a VM with the additional custom buttons *analyze* and *ping* defined:

```

{
  "href": "https://172.16.30.129/api/vms/173",
  "id": "173",
  "vendor": "vmware",
  "name": "aab-ldap",
  "location": "aab-ldap/aab-ldap.vmx",
  "host_id": "4",
  "created_on": "2018-01-09T20:23:37Z",
  "updated_on": "2018-01-09T20:23:37Z",
  "storage_id": "8",
  "guid": "dee6d969-7c69-44b6-bcd3-3b14f21702c5",
  "ems_id": "1",
  "uid_ems": "420c8f0d-4a8f-0b6d-fc3a-0c3cfab7b33f",
  "tools_status": "toolsNotRunning",
  "standby_action": "checkpoint",
  "power_state": "off",
  ...
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "add_event",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    ...
    {
      "name": "set_miq_server",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
    {
      "name": "scan",
      "method": "post",
      "href": "https://172.16.30.129/api/vms/173"
    },
  ],
}

```

```
{
  {
    "name": "delete",
    "method": "delete",
    "href": "https://172.16.30.129/api/vms/173"
  },
  {
    "name": "analyze",
    "method": "post",
    "href": "https://172.16.30.129/api/vms/173"
  },
  {
    "name": "ping",
    "method": "post",
    "href": "https://172.16.30.129/api/vms/173"
  }
}
]
```

Here, the *analyze* and *ping* are custom buttons for the VM and are returned after the common actions.

For example, to request the *analyze* custom action shown above, this is simply a POST action as follows:

Request:

POST /api/vms/173

```
{
  "action" : "analyze"
}
```

Response:

Responses from custom actions are the normal action responses from the REST API:

```
{
  "success": true,
  "message": "Invoked custom action analyze for VM id: 173",
  "href": "http://localhost:3000/api/vms/173"
}
```

- [Custom Attributes](#)
 - [Querying Custom Attributes](#)
 - [Specifying Attribute Actions](#)
 - [Custom Attribute Actions](#)
 - [Adding Custom Attributes](#)
 - [Editing Custom Attributes](#)
 - [Deleting Custom Attributes](#)

Custom Attributes

Custom Attributes can be managed for several types of resources. This is done via the `custom_attributes` sub-collection as follows:

Resource Type	Custom Attribute Subcollection
Virtual Machines	/api/vms/:id/custom_attributes
Cloud Instances	/api/instances/:id/custom_attributes
Providers	/api/providers/:id/custom_attributes

Resource Type	Custom Attribute Subcollection
Services	/api/services/:id/custom_attributes

- [Querying Custom Attributes](#)
- [Specifying Custom Attributes](#)
- [Custom Attribute Actions](#)
- [Adding Custom Attributes](#)
- [Editing Custom Attributes](#)
- [Deleting Custom Attributes](#)

Querying Custom Attributes

Custom attributes of a resource can be asked for by expanding the subcollection as follows:

```
GET /api/vms/:id?expand=custom_attributes
GET /api/instances/:id?expand=custom_attributes
GET /api/providers/:id?expand=custom_attributes
GET /api/services/:id?expand=custom_attributes
```

Optionally, just the custom attributes of a resource can be queried:

```
GET /api/vms/:id/custom_attributes
GET /api/instances/:id/custom_attributes
GET /api/providers/:id/custom_attributes
GET /api/services/:id/custom_attributes
```

Specifying Attribute Actions

Custom attributes are specified as follows:

```
{
  "name" : "...",
  "value" : "...",
  "field_type" : "...",
  "source" : "...",
  "section" : "...",
}
```

For adding custom attributes, **name** field is required.

field_type, **source** and **section** are optional.

There is no default value for **field_type**. Allowed values for **field_type** are **DateTime**, **Date** and **Time**.

Default value for **source** is **EVM**. If **source** is set to **VC**, the custom attribute field will be updated on the provider.

Default value for **section** is **metadata**. Allowed values for **section** include **metadata** and **cluster_settings**.

For editing a custom attribute, specify the custom attribute either by **name**:

```
{
  "name" : "...",
  "value" : "...",
}
```

Or by href:

```
{
  "href" : "http://localhost:3000/api/vms/:id/custom_attributes/:id",
  "value" : "...",
}
```

Custom Attribute Actions

Action	Description
add	Adding Custom Attributes to a Resource
edit	Editing Custom Attributes of a Resource
delete	Deleting Custom Attributes from a Resource

Adding Custom Attributes

Adding custom attributes to a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes
```

```
{
  "action" : "add",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

For example, adding two custom attributes:

```
{
  "action" : "add",
  "resources" : [
    { "name" : "ca1", "value" : "value 1" },
    { "name" : "ca2", "value" : "value 2" }
  ]
}
```

Editing Custom Attributes

Editing custom attributes of a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes
```

```
{
  "action" : "edit",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

For example:

```
{
  "action" : "edit",
  "resources" : [
    { "name" : "ca_name1", "value" : "updated custom attribute value 1" },
    { "name" : "ca_name2", "value" : "updated custom attribute value 2" }
  ]
}
```

Deleting Custom Attributes

Deleting custom attributes of a Resource:

```
POST /api/vms/:id/custom_attributes
POST /api/instances/:id/custom_attributes
POST /api/providers/:id/custom_attributes
POST /api/services/:id/custom_attributes
```

```
{
  "action" : "delete",
  "resources" : [
    { ... },
    { ... },
    ...
  ]
}
```

In the following example, a request to delete custom attributes of a VM by name and by href:

```
{
  "action" : "delete",
  "resources" : [
    { "name" : "ca_name1" },
    { "href" : "http://localhost:3000/api/vms/:id/custom_attributes/:id" }
  ]
}
```

Custom attributes can be deleted via a *delete* action as follows:

```
POST /api/vms/:id/custom_attributes/:ca_id
POST /api/instances/:id/custom_attributes/:ca_id
POST /api/providers/:id/custom_attributes/:ca_id
POST /api/services/:id/custom_attributes/:ca_id

{
  "action" : "delete"
}
```

They can also be deleted via the DELETE method:

```
DELETE /api/vms/:id/custom_attributes/:ca_id
DELETE /api/instances/:id/custom_attributes/:ca_id
DELETE /api/providers/:id/custom_attributes/:ca_id
DELETE /api/services/:id/custom_attributes/:ca_id
```

- [Custom Button Management](#)
 - [Managing Custom Button Sets](#)
 - [Querying Custom Button Sets](#)
 - [Creating Custom Button Sets](#)
 - [Editing Custom Button Sets](#)
 - [Deleting Custom Button Sets](#)
 - [Managing Custom Buttons](#)
 - [Querying Custom Buttons](#)

- [Creating Custom Buttons](#)
- [Editing Custom Buttons](#)
- [Deleting Custom Buttons](#)

Custom Button Management

Management of Custom Buttons is provided with the following collections:

`/api/custom_button_sets`
`/api/custom_buttons`

- [Managing Custom Button Sets](#)
 - [Querying Custom Button Sets](#)
 - [Creating Custom Button Sets](#)
 - [Editing Custom Button Sets](#)
 - [Deleting Custom Button Sets](#)
- [Managing Custom Buttons](#)
 - [Querying Custom Buttons](#)
 - [Creating Custom Buttons](#)
 - [Editing Custom Buttons](#)
 - [Deleting Custom Buttons](#)

Managing Custom Button Sets

Querying Custom Button Sets

Querying all available Custom Button Sets:

`GET /api/custom_button_sets`

Getting details on a specific Custom Button Set:

`GET /api/custom_button_sets/:id`

Creating Custom Button Sets

Custom Button Sets can be created as follows:

`POST /api/custom_button_sets`

```
{
  "name"      : "Generic Object Custom Button Group",
  "description": "Generic Object Custom Button Group Description",
  "set_data"   : {
    "button_icon"      : "ff ff-view-expanded",
    "button_color"     : "#4727ff",
    "display"          : true,
    "applies_to_class" : "GenericObjectDefinition",
    "applies_to_id"    : "50"
  }
}
```



```
}  
}
```

Editing Custom Button Sets

Updating Custom Button Sets can be done via the *edit* action as follows:

```
POST /api/custom_button_sets/:id
```

```
{  
  "action" : "edit",  
  "resource" : {  
    "name" : "Updated Custom Button Group",  
    "description" : "Updated Custom Button Group Description",  
    "set_data" : {  
      "button_icon" : "ff ff-closed"  
    }  
  }  
}
```

Custom Button Sets can also be updated in bulk:

```
POST /api/custom_button_sets
```

```
{  
  "action" : "edit",  
  "resources" : [  
    { "id" : "11", "set_data" : { "button_icon" : "ff ff-closed" } },  
    { "id" : "12", "set_data" : { "button_icon" : "ff ff-closed" } },  
    { "id" : "13", "set_data" : { "button_icon" : "ff ff-closed" } }  
  ]  
}
```

Custom button sets can also be updated with the PUT method as follows:

```
PUT /api/custom_button_sets/:id
```

```
{  
  "name" : "Updated Custom Button Group",  
  "description" : "Updated Custom Button Group Description",  
  "set_data" : {  
    "button_icon" : "ff ff-closed"  
  }  
}
```

Deleting Custom Button Sets

Custom Button Sets can be deleted via the *delete* action as follows:

```
POST /api/custom_button_sets/:id
```

```
{  
  "action" : "delete"  
}
```

and in bulk as follows:

```
POST /api/custom_button_sets
```

```
{  
  "action" : "delete",  
  "resources" : [  
    { "href" : "http://localhost:3000/api/custom_button_sets/21" },  
  ]  
}
```

```

    { "href" : "http://localhost:3000/api/custom_button_sets/22" },
    { "id" : "23" }
  ]
}

```

A single Custom Button Set can also be deleted with the DELETE method as follows:

```
DELETE /api/custom_button_sets/:id
```

Managing Custom Buttons

Querying Custom Buttons

Querying all available Custom Buttons:

```
GET /api/custom_buttons
```

Getting details on a specific Custom Button:

```
GET /api/custom_buttons/:id
```

Creating Custom Buttons

Custom Buttons can be created as follows:

```
POST /api/custom_buttons
```

```

{
  "name"          : "Generic Object Custom Button",
  "description"   : "Generic Object Custom Button Description",
  "applies_to_class" : "GenericObjectDefinition",
  "options"       : {
    "button_icon"   : "ff ff-view-expanded",
    "button_color"  : "#4727ff",
    "display"       : true
  },
  "resource_action" : {
    "ae_namespace"  : "SYSTEM",
    "ae_class"      : "PROCESS",
    "dialog_id"     : "1"
  },
  "visibility"     : {
    "roles"         : ["_ALL_"]
  }
}

```

Editing Custom Buttons

Updating Custom Buttons can be done via the *edit* action as follows:

```
POST /api/custom_buttons/:id
```

```

{
  "action" : "edit",
  "resource" : {
    "name" : "Updated Custom Button",
    "description" : "Updated Custom Button Description",
    "resource_action" : {
      "ae_namespace" : "SYSTEM2"
    }
  }
}

```

Custom Buttons can also be updated in Bulk as follows:

POST /api/custom_buttons

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/custom_buttons/31",
      "options" : { "display" : false }
    },
    {
      "href" : "http://localhost:3000/api/custom_buttons/32",
      "options" : { "display" : false }
    },
    {
      "id" : "33",
      "options" : { "display" : false }
    }
  ]
}
```

Deleting Custom Buttons

Custom Buttons can be deleted via the *delete* action as follows:

POST /api/custom_buttons/:id

```
{
  "action" : "delete"
}
```

Multiple Custom Buttons can also be deleted in a single request as follows:

POST /api/custom_buttons

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/custom_buttons/21" },
    { "id" : "22" },
    { "id" : "23" }
  ]
}
```

Single Custom Buttons can also be deleted with the DELETE method as follows:

DELETE /api/custom_buttons/:id

- [Event Streams](#)
 - [Filtering by event type](#)
 - [Filtering by timestamp](#)
 - [Filtering by target type](#)
 - [Filtering by target type and id](#)

Event Streams

Querying of Event Streams is made available via the following collection:

/api/event_streams

Event Streams can be queried as follows:

```
GET /api/event_streams
```

An individual Event Stream can also be fetched as follows:

```
GET /api/event_streams/:id
```

Relevant filter[] parameters when Event Streams are queried:

Parameter	Examples
event_type	event_type=vm_start, event_type=vm_stop
timestamp	timestamp>2017-01-03, timestamp<2017-12-31
target_type	target_type=VmOrTemplate
target_id	target_id=11

Filtering by event type

```
GET /api/event_streams?filter[]=event_type=vm_start
```

Filtering by timestamp

```
GET /api/event_streams?filter[]=timestamp>2017-01-03
&filter[]=timestamp<2017-01-10
```

Filtering by target type

```
GET /api/event_streams?filter[]=target_type=VmOrTemplate
```

Filtering by target type and id

```
GET /api/event_streams?filter[]=target_type=VmOrTemplate
&filter[]=target_id=21
```

- [Flavor Management](#)
 - [Querying Flavors](#)
 - [Managing Flavors](#)
 - [Create Flavors](#)
 - [Delete Flavors](#)

Flavor Management

Management of Flavors is available via the following endpoint:

```
/api/flavors
```

Querying Flavors

Querying all available Flavors:

```
GET /api/flavors
```

Getting details on a specific Flavor:

```
GET /api/flavors/:id
```

Querying the Flavors for a specific Provider:

```
GET /api/providers/:id/flavors
```

Getting details on a specific Flavor of a Provider:

```
GET /api/providers/:id/flavors/:flavor_id
```

Managing Flavors

The following actions are available on Flavors of Providers:

Action	Description
create	Create Flavors
delete	Delete Flavors

Create Flavors

Flavors can be created directly for a Provider as follows:

```
POST /api/providers/:id/flavors

{
  "name" : "base-system",
  "cpu_cores" : 2,
  "memory" : 4096000000,
  "root_disk_size" : 16384000000
}
```

Delete Flavors

Flavors of a Provider can be deleted via the *delete* action:

```
POST /api/providers/:id/flavors/:flavor_id

{
  "action" : "delete"
}
```

Multiple Flavors can also be deleted in a single request as follows:

```
POST /api/providers/:id/flavors

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/flavors/6" },
    { "id" : "7" }
  ]
}
```

Single Flavors can also be deleted with the DELETE HTTP method as follows:

```
DELETE /api/providers/:id/flavors/:flavor_id
```

- [Generic Object Management](#)
 - [Generic Object Definitions](#)
 - [Querying Generic Object Definitions](#)
 - [Creating Generic Object Definitions](#)
 - [Editing Generic Object Definitions](#)
 - [Add Attributes](#)
 - [Remove Attributes](#)

- [Add Methods](#)
- [Remove Methods](#)
- [Add Associations](#)
- [Remove Associations](#)
- [Deleting Generic Object Definitions](#)
- [Generic Objects](#)
 - [Querying Generic Objects](#)
 - [Request:](#)
 - [Response:](#)
 - [Creating Generic Objects](#)
 - [Editing Generic Objects](#)
 - [Calling Generic Object Methods](#)
 - [Deleting Generic Objects](#)

Generic Object Management

Management of Generic Objects is provided with the following collections:

`/api/generic_object_definitions`
`/api/generic_objects`

- [Managing Generic Object Definitions](#)
 - [Querying Generic Object Definitions](#)
 - [Creating Generic Object Definitions](#)
 - [Editing Generic Object Definitions](#)
 - [Deleting Generic Object Definitions](#)
- [Managing Generic Objects](#)
 - [Querying Generic Objects](#)
 - [Creating Generic Objects](#)
 - [Editing Generic Objects](#)
 - [Calling Generic Object Methods](#)
 - [Deleting Generic Objects](#)

Generic Object Definitions

Querying Generic Object Definitions

Querying all available Generic Object Definitions:

GET `/api/generic_object_definitions`

Getting details on a specific Generic Object Definition:

GET `/api/generic_object_definitions/:id`

Getting details on a specific Generic Object Definition by name:

```
GET /api/generic_object_definitions/:name
```

Querying all the Generic Objects defined for a specific Generic Object Definition:

```
GET /api/generic_object_definitions/:id/generic_objects
```

Getting a specific Generic Object defined for a Generic Object Definition:

```
GET /api/generic_object_definitions/:id/generic_objects/:generic_object_id
```

Fetching the vms associations for a specific Generic Object:

```
GET /api/generic_object_definitions/:id/generic_objects/:generic_object_id?
associations=vms
```

Creating Generic Object Definitions

Generic Object Definitions can be created as follows:

```
POST /api/generic_object_definitions
```

```
{
  "name"      : "LoadBalancer",
  "description" : "LoadBalancer description",
  "properties" : {
    "attributes" : {
      "address"      : "string",
      "last_restart" : "datetime"
    },
    "associations" : {
      "vms"          : "Vm",
      "services"     : "Service"
    },
    "methods" : [
      "add_vm",
      "remove_vm"
    ]
  },
  "picture" : {
    "extension" : "png",
    "content"   : "iVBORw0KGgoAAAANSUHEUgAAAAEAAAABCAyAAAAfFcSJAABBGdBTUEAALGP\"\\
      \"C/xhBQAAACBjSFJNAAB6JgAAgIQAAPoAAACA6AAAAdTAAAOpgAAA6mAAAF3Cc\"\\
      \"ule8AAAACXBIWXMAAAsTAAALEwEAmpwYAAABWWlUWHRYTUw6Y29tLmFkb2Jl\"\\
      \"LnhtcAAAAAAAPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyIg\"\\
      \"eDp4bXB0az0iWE1QIENvcmluZS40LjAiPgogICA8cmRmOlJERiB4bWxuczpy\"\\
      \"ZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1u\"\\
      \"cyMiPgogICA8cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0iIgorICA8\"\\
      \"ICAgICAgICB4bWxuczp0aWZmPSJodHRwOi8vbnMuYWRvYm9uY29tL3R5bWVv\"\\
      \"MS4wLyI+CiAgICAgICAgIDx0aWZmOk9yaWVudGF0aW9uPjE8L3R5bWV6T3Jp\"\\
      \"ZW50YXRpb24+CiAgICAgICAgIDwvcmluZS40LjN5aXB0aW9uPgogICA8L3JkZjps\"\\
      \"REY+CjwveDp4bXBtZXRhPgpmwidZAAAADULEQVQIHWNgyGcWbQAAQGA+3N0+\"\\
      \"xQAAAABJR5ErkJggg==\"
  }
}
```

Example of creating a Generic Object Definition with a picture reference:

```
{
  "name"      : "LoadBalancer",
  "description" : "LoadBalancer description",
  "properties" : {
    "attributes" : {
      "address"      : "string",
```

```

    "last_restart" : "datetime"
  },
  "associations" : {
    "vms" : "Vm",
    "services" : "Service"
  },
  "methods" : [
    "add_vm",
    "remove_vm"
  ]
},
"picture" : {
  "href" : "http://localhost:3000/api/pictures/11"
}
}

```

Note: picture references can be by *href* or *id*.

Editing Generic Object Definitions

Updating Generic Object Definitions can be done via the *edit* action as follows:

POST /api/generic_object_definitions/:id

or by name:

POST /api/generic_object_definitions/:name

```

{
  "action" : "edit",
  "resource" : {
    "description" : "Latest LoadBalancer Description"
  }
}

```

Generic object definitions can also be updated with the PUT method as follows:

PUT /api/generic_object_definitions/:id

```

{
  "name" : "New LoadBalancer Name",
  "description" : "New LoadBalancer Description"
}

```

Pictures can be removed from Generic Object Definitions by sending in an empty picture reference via any of the edit methods, for example:

POST /api/generic_object_definitions/:name

```

{
  "action" : "edit",
  "resource" : {
    "picture" : {}
  }
}

```

Generic Object Definitions can be edited by href, id or name in bulk as in the following example:

POST /api/generic_object_definitions

```

{
  "action" : "edit",
  "resources" : [

```



```

{
  "name" : "Load Balancer",
  "resource" : {
    "name" : "Enhanced Load Balancer",
    "description" : "Description for Enhanced Load Balancer",
    "picture" : { "id" : "16" }
  }
},
{
  "href" : "http://localhost:3000/api/generic_object_definitions/12",
  "resource" : {
    "name" : "NetworkScanner",
    "description" : "Description for Network Scanner"
  }
},
{
  "id" : "13",
  "resource" : {
    "name" : "NetworkFirewall",
    "description" : "Description for Network Firewall"
  }
}
]
}

```

Editing Generic Object Definitions can also be done by using one of the following actions for updating *attributes*, *methods* and *associations*:

- [add_attributes](#)
- [remove_attributes](#)
- [add_methods](#)
- [remove_methods](#)
- [add_associations](#)
- [remove_associations](#)

Add Attributes

POST /api/generic_object_definitions/:id

```

{
  "action" : "add_attributes",
  "resource" : {
    "attributes" : {
      "optional_attribute1" : "string",
      "optional_attribute2" : "boolean"
    }
  }
}

```

Remove Attributes

POST /api/generic_object_definitions/:id

```

{
  "action" : "remove_attributes",
  "resource" : {
    "attributes" : {

```

```

    "unused_attribute9" : "string"
  }
}
}

```

Add Methods

```

POST /api/generic_object_definitions/:id

{
  "action" : "add_methods",
  "resource" : {
    "methods" : [ "go_method1", "go_method2" ]
  }
}

```

Remove Methods

```

POST /api/generic_object_definitions/:id

{
  "action" : "remove_methods",
  "resource" : {
    "methods" : [ "go_method3" ]
  }
}

```

Add Associations

```

POST /api/generic_object_definitions/:id

{
  "action" : "add_associations",
  "resource" : {
    "associations" : {
      "az" : "AvailabilityZone",
      "chargeback" : "ChargebackVm"
    }
  }
}

```

Remove Associations

```

POST /api/generic_object_definitions/:id

{
  "action" : "remove_associations",
  "resource" : {
    "associations" : {
      "chargeback" : "ChargebackVm"
    }
  }
}

```

Deleting Generic Object Definitions

Generic Object Definitions can be deleted via the *delete* action as follows:

```

POST /api/generic_object_definitions/:id

```

or by name:

```
POST /api/generic_object_definitions/:name
```

```
{  
  "action" : "delete"  
}
```

Multiple Generic Object Definitions can also be deleted in a single request as follows:

```
POST /api/generic_object_definitions
```

```
{  
  "action" : "delete",  
  "resources" : [  
    { "href" : "http://localhost:3000/api/generic_object_definitions/11" },  
    { "id" : "12" }  
  ]  
}
```

Single Generic Object Definitions can also be deleted with the DELETE method as follows:

```
DELETE /api/generic_object_definitions/:id
```

or by name:

```
DELETE /api/generic_object_definitions/:name
```

Generic Objects

Querying Generic Objects

Querying all available Generic Objects:

```
GET /api/generic_objects
```

Getting details on a specific Generic Object:

```
GET /api/generic_objects/:id
```

Querying all the Generic Objects defined for a specific Service:

```
GET /api/services/:id/generic_objects
```

Querying Generic Objects and related picture and image_href for a specific Service:

Request:

```
GET /api/services/:id?expand=generic_objects&attributes=generic_objects.picture
```

Response:

```
{  
  "href" : "http://localhost:3000/api/services/645",  
  "id" : "645",  
  "name" : "Test Azure Service",  
  "description" : "Azure",  
  "guid" : "6ce8f6c2-7f93-4a7e-b618-da9acab470ac",  
  "options": { },  
  "display": false,  
  "created_at" : "2017-08-07T16:52:46Z",  
  "updated_at" : "2017-08-07T16:52:46Z",  
  "miq_group_id" : "1",  
  "retired": false,  
}
```

```

"tenant_id" : "1",
"initiator" : "user",
"generic_objects": [
  {
    "href" : "http://localhost:3000/api/services/645/generic_objects/1",
    "id" : "1",
    "name" : "generic_object_1",
    "generic_object_definition_id" : "1",
    "created_at" : "2017-08-15T17:35:08Z",
    "updated_at" : "2017-09-07T12:05:14Z",
    "properties": {
      "vms" : [ 615 ],
      "start" : true
    },
    "picture" : {
      "href" : "http://localhost:3000/api/picture/1",
      "id" : "1",
      "resource_id" : "1",
      "resource_type" : "GenericObjectDefinition",
      "image_href" : "http://localhost:3000/pictures/1.png",
      "extension" : "png"
    }
  }
],
"actions" : {
  ...
}
}

```

Creating Generic Objects

Generic Objects can be created as follows:

POST /api/generic_objects

```

{
  "generic_object_definition" : { "href" :
"http://localhost:3000/api/generic_object_definitions/11" },
  "name" : "gol_name",
  "uid" : "optional_uid",
  "property_attributes" : {
    "widget" : "widget value",
    "supports_feature_a" : false
  },
  "associations" : {
    "vms" : [
      { "href" : "http://localhost:3000/api/vms/11" },
      { "id" : "12" }
    ]
  }
}

```

Generic Objects can also be directly created for a specific Generic Object Definition type as follows:

POST /api/generic_object_definitions/:id/generic_objects

```

{
  "name" : "gol_name",
  "uid" : "optional_uid",
  "property_attributes" : {
    "widget" : "widget value",
    "supports_feature_a" : false
  },
}

```

```

"associations" : {
  "vms" : [
    { "href" : "http://localhost:3000/api/vms/11" },
    { "id" : "12" }
  ]
}
}

```

In which case, the `generic_object_definition` reference is not needed.

Editing Generic Objects

Updating Generic Objects can be done via the *edit* action as follows:

POST `/api/generic_objects/:id`

```

{
  "action" : "edit",
  "resource" : {
    "name" : "updated_name",
    "property_attributes" : {
      "widget" : "updated_widget"
    }
  }
}

```

Calling Generic Object Methods

When generic objects have methods defined in their definitions, those methods are exposed with the generic objects as actions.

The following example shows a generic object with two methods *method_a* and *method_b*:

GET `/api/generic_objects/101`

```

{
  "href" : "http://localhost:3000/api/generic_objects/101",
  "id" : "101",
  "name" : "sample_object",
  "generic_object_definition_id" : "5",
  "property_attributes" : {
    "widget" : "a sample widget",
    "object_flag" : true
  },
  "actions" : [
    {
      "name" : "edit",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "delete",
      "method" : "post",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "delete",
      "method" : "delete",
      "href" : "http://localhost:3000/api/generic_objects/101"
    },
    {
      "name" : "method_a",

```

```

    "method" : "post",
    "href" : "http://localhost:3000/api/generic_objects/101"
  },
  {
    "name" : "method_b",
    "method" : "post",
    "href" : "http://localhost:3000/api/generic_objects/101"
  }
]
}

```

With the above sample generic object, the custom methods can be invoked like any other action as follows:

POST /api/generic_objects/101

```

{
  "action" : "method_a",
  "resource" : {
    "attr1" : "value 1",
    "attr2" : "value 2"
  }
}

```

Deleting Generic Objects

Generic Objects can be deleted via the *delete* action as follows:

POST /api/generic_objects/:id

```

{
  "action" : "delete"
}

```

Multiple Generic Objects can also be deleted in a single request as follows:

POST /api/generic_objects

```

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/generic_objects/21" },
    { "id" : "22" },
    { "id" : "23" }
  ]
}

```

Single Generic Objects can also be deleted with the DELETE method as follows:

DELETE /api/generic_objects/:id

- [Group Management](#)
 - [Querying Groups](#)
 - [Creating Groups](#)
 - [Editing Groups](#)
 - [Deleting Groups](#)

Group Management

Management of User Groups is provided via the following collection:

/api/groups

Full CRUD actions on groups is available:

- [Querying Groups](#)
- [Creating Groups](#)
- [Editing Groups](#)
- [Deleting Groups](#)

Querying Groups

Querying all groups in the system:

```
GET /api/groups
```

Getting details on a specific group:

```
GET /api/groups/:id
```

Creating Groups

Groups can be created via a POST to the groups collection or via the **create** action signature which also allows creation of multiple groups in a single request.

```
POST /api/groups
```

```
{
  "description" : "sample_group",
  "role" : { "id" : "2" },
  "tenant" : { "href" : "http://localhost:3000/api/tenants/1" },
  "filters" : {
    "belongsto" : [ "/managed/area/1", "/managed/area/2" ],
    "managed" : [ [ "/managed/infra/1", "/managed/infra/2" ] ]
  }
}
```

Note:

role can be specified via **name**, **href**, or **id**

tenant can be specified via **href** or **id**

or creating multiple groups:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "sample_group1", ... },
    { "description" : "sample_group2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Groups.

Editing Groups

```
POST /api/groups/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "updated_sample_group",
    "filters" : {
      "belongsto" : [ "/managed/area/1", "/managed/area/2", "/managed/area/3" ],
      "managed" : [[ "/managed/infra/1", "/managed/infra/2"],
["/managed/other/3"]]
    }
  }
}
```

or editing multiple groups:

POST /api/groups

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/groups/101",
      "description" : "updated_sample_group1"
    },
    {
      "href" : "http://localhost:3000/api/groups/102",
      "description" : "updated_sample_group2"
    },
    ...
  ]
}
```

Deleting Groups

groups can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/groups/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/groups/101

Deleting multiple groups can be done as follows:

POST /api/groups

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/groups/101" },
    { "href" : "http://localhost:3000/api/groups/102" },
    ...
  ]
}
```

- [Host Management](#)
 - [Querying Hosts](#)
 - [Editing a Host](#)

Host Management

Management of Hosts adds support of the following actions:

Action	Description
edit	Edit a Host

- [Querying Hosts](#)
- [Editing a Host](#)

Querying Hosts

Hosts are queried via the primary collection URL:

`/api/hosts`

Editing a Host

Currently, only the host password can be edited via the API. All other edit requests will be rejected.

The host password is provided by a *credentials* hash similar to the signature supported for providers.

POST `/api/hosts/:id`

```
{
  "action" : "edit",
  "resource" : {
    "credentials" : {
      "password" : "updated_password"
    }
  }
}
```

When not specified, the authentication type used for the password is *default*:

POST `/api/hosts/:id`

```
{
  "action" : "edit",
  "resource" : {
    "credentials" : {
      "auth_type" : "default",
      "password" : "updated_password"
    }
  }
}
```

Password for multiple hosts can also be updated with a single request as follows:

POST `/api/hosts`

```
{
  "action" : "edit",
  "resources" : [
    { "id" : "101", "credentials" : { "password" : "updated_password" } },
    { "id" : "102", "credentials" : { "password" : "updated_password" } },
    { "id" : "103", "credentials" : { "password" : "updated_password" } },
    ...
  ]
}
```

Note:

host can be referenced by **href** or **id**

- [Instance Management](#)
 - [Targeting Instances](#)
 - [Querying Instances](#)
 - [Load Balancers](#)
 - [Starting an Instance](#)
 - [Stopping an Instance](#)
 - [Suspending an Instance](#)
 - [Pausing an Instance](#)
 - [Terminating an Instance](#)
 - [Shelving an Instance](#)
 - [Rebooting Guest in an Instance](#)
 - [Resetting an Instance](#)

Instance Management

Management of instances adds support for the following actions:

Action	Description
start	Starts an Instance
stop	Stops an Instance
suspend	Suspending an Instance
pause	Pausing an Instance
reboot_guest	Rebooting Guest in an Instance
reset	Resetting an Instance
shelve	Shelving an Instance
terminate	Terminating an Instance

- [Targeting Instances](#)
- [Querying Instances](#)
- [Starting an Instance](#)
- [Stopping an Instance](#)
- [Suspending an Instance](#)
- [Pausing an Instance](#)
- [Rebooting Guest in an Instance](#)
- [Resetting an Instance](#)
- [Shelving an Instance](#)
- [Terminating an Instance](#)

Targeting Instances

These actions can be triggered on individual instance resources:

```
/api/instances/:id
```

As simply as POSTing the following action to an instance.

```
{  
  "action" : "start"  
}
```

Requests can also be made on multiple instances by targeting the primary collection:

```
/api/instances
```

```
{  
  "action" : "start",  
  "resources" : [  
    { "href" : "http://localhost:3000/api/instances/101" },  
    { "href" : "http://localhost:3000/api/instances/102" },  
    ...  
  ]  
}
```

Querying Instances

Instances are queried via the primary collection URL:

```
/api/instances
```

Filtering, sorting and paging as mentioned on the [Querying](#) page.

Load Balancers

Load Balancers of an instance can be queried via the load_balancers subcollection as follows:

Query all Load Balancers of an instance:

```
GET /api/instances/:id/load_balancers
```

Query a specific Load Balancer of an instance:

```
GET /api/instances/:id/load_balancers/:load_balancer_id
```

Or expanding the Load Balancers for the specific instance as follows:

```
GET /api/instances/:id?expand=load_balancers
```

Starting an Instance

```
{  
  "action" : "start"  
}
```

Stopping an Instance

```
{  
  "action" : "stop"  
}
```

Suspending an Instance

```
{  
  "action" : "suspend"  
}
```

Pausing an Instance

```
{  
  "action" : "pause"  
}
```

Terminating an Instance

```
{  
  "action" : "terminate"  
}
```

Shelving an Instance

```
{  
  "action" : "shelve"  
}
```

Rebooting Guest in an Instance

```
{  
  "action" : "reboot-guest"  
}
```

Resetting an Instance

```
{  
  "action" : "reset"  
}
```

- [Metric Rollups](#)
 - [Getting 60 Daily Metric Rollups](#)
 - [Getting Hourly Metric Rollups for a range of Dates](#)
 - [Getting Hourly Metric Rollups for a Service](#)
 - [Getting 30 Daily Metric Rollups for a Vm](#)

Metric Rollups

Querying of Metric Rollups is made available via the following collection:

`/api/metric_rollups`

Supported query parameters:

Parameter	Examples
resource_type	VmOrTemplate, Service or any resource type that supports Metric Rollups
resource_ids[]	id1
resource_ids[]	id2
resource_ids[]	...
capture_interval	hourly, daily
start_date	2018-01-10, 2018-01-10T11:20:00Z
end_date	2018-01-17, 2017-12-31T23:59:00Z

Getting 60 Daily Metric Rollups

```
GET /api/metric_rollups?resource_type=VmOrTemplate
    &resource_ids[]=101
    &resource_ids[]=102
    &resource_ids[]=103
    &capture_interval=daily
    &start_date='2017-10-01'
    &limit=60
```

Getting Hourly Metric Rollups for a range of Dates

```
GET /api/metric_rollups?resource_type=Service
    &resource_ids[]=55
    &resource_ids[]=56
    &capture_interval=hourly
    &start_date='2017-11-15'
    &end_date='2017-11-18'
```

Metric rollups can also be obtained for a specific resource as a subcollection:

```
/api/vms/:id/metric_rollups
```

```
/api/services/:id/metric_rollups
```

Getting Hourly Metric Rollups for a Service

```
GET /api/services/57/metric_rollups?capture_interval=hourly&start_date='2017-06-15'
```

Getting 30 Daily Metric Rollups for a Vm

```
GET /api/vms/104/metric_rollups?capture_interval=daily&start_date='2017-01-01'&limit=30
```

- [Network Routers](#)
 - [Querying Network Routers](#)
 - [Deleting Network Routers](#)
 - [Request:](#)
 - [Response:](#)

Network Routers

Management of Network Routers is provided via the following collection:

```
/api/network_routers
```

The following actions are available on Network Routers:

- [Querying Network Routers](#)
- [Deleting Network Routers](#)

Querying Network Routers

Querying all Network Routers in the system is simply:

```
GET /api/network_routers
```

Getting details on a specific Network Router:

```
GET /api/network_routers/:id
```

Deleting Network Routers

Network Routers can be deleted via either the **delete** POST action or via the DELETE HTTP method.

Request:

```
POST /api/network_routers/101
```

```
{
  "action" : "delete"
}
```

Response:

```
{
  "success" : true,
  "message" : "Deleting Networking Router id:101 name: 'test_network_router'",
  "task_id" : "33",
  "task_href" : "http://localhost:3000/api/tasks/33"
}
```

or simply:

```
DELETE /api/network_routers/101
```

Deleting multiple Network Routers can be done as follows:

```
POST /api/network_routers
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/network_routers/101" },
    { "href" : "http://localhost:3000/api/network_routers/102" },
    ...
  ]
}
```

- [Notifications Support](#)
- [Querying Notifications](#)
- [Marking Notifications as Seen](#)
 - [Deleting Notifications](#)

Notifications Support

Support for Notifications to the authenticated user is provided via the following collection:

```
/api/notifications
```

- [Querying Notifications](#)
- [Marking Notifications as Seen](#)
- [Deleting Notifications](#)

Querying Notifications

The following fetches the current notifications for the authenticated user:

GET /api/notifications

```
{
  "name": "notifications",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/notifications/44"
    },
    {
      "href": "http://localhost:3000/api/notifications/45"
    },
    {
      "href": "http://localhost:3000/api/notifications/46"
    }
  ],
  "actions": [
    {
      "name": "mark_as_seen",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    }
  ]
}
```

Or fetching additional details on the notifications as follows:

GET /api/notifications?expand=resources&attributes=details

```
{
  "name": "notifications",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/notifications/44",
      "id": "44",
      "user_id": "1",
      "seen": false,
      "details": {
        "level": "info",
        "created_at": "2016-11-03T17:23:29Z",
        "text": "Vm [aab-test] was powered on",
        "bindings": {
          "initiator": {
            "link": {
              "id": "1",
              "model": "User"
            },
            "text": "Administrator"
          }
        },
        "seen": false
      },
      "actions": [
        {
          "name": "mark_as_seen",
          "method": "post",

```

```

    "href": "http://localhost:3000/api/notifications/44"
  },
  {
    "name": "delete",
    "method": "post",
    "href": "http://localhost:3000/api/notifications/44"
  },
  {
    "name": "delete",
    "method": "delete",
    "href": "http://localhost:3000/api/notifications/44"
  }
]
},
{
  "href": "http://localhost:3000/api/notifications/45",
  "id": "45",
  "notification_id": "22",
  "user_id": "1",
  "seen": true,
  "details": {
    "level": "info",
    "created_at": "2016-11-03T17:23:29Z",
    "text": "Vm [aab-test2] was powered on",
    "bindings": {
      "initiator": {
        "link": {
          "id": "1",
          "model": "User"
        },
        "text": "Administrator"
      }
    }
  },
  "seen": true
},
"actions": [
  {
    "name": "mark_as_seen",
    "method": "post",
    "href": "http://localhost:3000/api/notifications/45"
  },
  {
    "name": "delete",
    "method": "post",
    "href": "http://localhost:3000/api/notifications/45"
  },
  {
    "name": "delete",
    "method": "delete",
    "href": "http://localhost:3000/api/notifications/45"
  }
]
},
{
  "href": "http://localhost:3000/api/notifications/46",
  "id": "46",
  "notification_id": "23",
  "user_id": "1",
  "seen": false,
  "details": {
    "level": "info",
    "created_at": "2016-11-03T17:23:54Z",
    "text": "Vm [aab-dev] was powered off",
    "bindings": {

```



```

        "initiator": {
          "link": {
            "id": "1",
            "model": "User"
          },
          "text": "Administrator"
        }
      },
      "seen": false
    },
    "actions": [
      {
        "name": "mark_as_seen",
        "method": "post",
        "href": "http://localhost:3000/api/notifications/46"
      },
      {
        "name": "delete",
        "method": "post",
        "href": "http://localhost:3000/api/notifications/46"
      },
      {
        "name": "delete",
        "method": "delete",
        "href": "http://localhost:3000/api/notifications/46"
      }
    ]
  },
  ],
  "actions": [
    {
      "name": "mark_as_seen",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    },
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/notifications"
    }
  ]
}

```

Marking Notifications as Seen

Marking Notifications as seen for the authenticated user is done via the **mark_as_seen** action as follows:

POST /api/notifications/:id

```

{
  "action" : "mark_as_seen"
}

```

Or, marking several notifications as seen in a single call as follows:

POST /api/notifications

```

{
  "action" : "mark_as_seen",
  "resources" : [
    { "href" : "http://localhost:3000/api/notifications/44" },
    { "href" : "http://localhost:3000/api/notifications/45" },

```

```
...
]
}
```

Deleting Notifications

Deleting a notification is available via the **delete** action or DELETE on the individual notification as follows:

POST /api/notifications/:id

```
{
  "action" : "delete"
}
```

Or simply:

DELETE /api/notifications/:id

Also, deleting multiple notifications is available via the **delete** action on the collection passing in references to the different notifications:

POST /api/notifications

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/notifications/44" },
    { "href" : "http://localhost:3000/api/notifications/45" },
    ...
  ]
}
```

- [Orchestration Support](#)
 - [Orchestration Templates ===](#)
 - [Querying Orchestration Templates](#)
 - [Creating Orchestration Templates](#)
 - [Editing Orchestration Templates](#)
 - [Copying Orchestration Templates](#)
 - [Deleting Orchestration Templates](#)

Orchestration Support

Support for orchestration is provided by the following:

- [Orchestration Templates](#)

Orchestration Templates ===

Management of orchestration templates is provided with the following collection:

/api/orchestration_templates

Full CRUD actions on orchestration templates is available:

- [Querying Orchestration Templates](#)
- [Creating Orchestration Templates](#)
- [Editing Orchestration Templates](#)

- [Copying Orchestration Templates](#)
- [Deleting Orchestration Templates](#)

Querying Orchestration Templates

Querying all orchestration templates in the system:

```
GET /api/orchestration_templates
```

Getting details on a specific orchestration template:

```
GET /api/orchestration_templates/:id
```

Creating Orchestration Templates

Orchestration templates can be created via a POST to the orchestration templates collection or via the **create** action signature which also allows creation of multiple orchestration templates in a single request.

```
POST /api/orchestration_templates
```

```
{
  "name" : "azure-single-vm-from-user-image",
  "type" : "OrchestrationTemplateAzure",
  "description" : "Create a single VM from a user image",
  "content" : " ... ",
  "orderable" : true,
  ...
}
```

or creating multiple orchestration templates:

```
{
  "action" : "create",
  "resources" : [
    { "description" : "sample_orchestration_template1", ... },
    { "description" : "sample_orchestration_template2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating orchestration templates.

Editing Orchestration Templates

```
POST /api/orchestration_templates/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "description" : "updated_sample_orchestration_template",
  }
}
```

or editing multiple orchestration templates:

```
POST /api/orchestration_templates
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/orchestration_templates/101",
      "description" : "updated_sample_orchestration_template1"
    },
    {
      "href" : "http://localhost:3000/api/orchestration_templates/102",
      "description" : "updated_sample_orchestration_template2"
    },
    ...
  ]
}
```

Copying Orchestration Templates

This capability allows orchestration templates to be copied onto new orchestration templates with updated content via the *copy* action as follows:

POST /api/orchestration_templates/:id

```
{
  "action" : "copy",
  "resource" : {
    "content" : "{ 'Description' : 'Copied Test Content' }"
  }
}
```

or copying multiple orchestration templates:

POST /api/orchestration_templates

```
{
  "action" : "copy",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/orchestration_templates/101",
      "content" : "{ 'Description' : 'Copied Test Content 1' }"
    },
    {
      "href" : "http://localhost:3000/api/orchestration_templates/102",
      "content" : "{ 'Description' : 'Copied Test Content 2' }"
    },
    ...
  ]
}
```

Deleting Orchestration Templates

Orchestration templates can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/orchestration_templates/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/orchestration_templates/101

Deleting multiple orchestration templates can be done as follows:

POST /api/orchestration_templates

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/orchestration_templates/101" },
    { "href" : "http://localhost:3000/api/orchestration_templates/102" },
    ...
  ]
}
```

- [Physical Server Management](#)
 - [Querying Physical Servers](#)
 - [Managing Physical Servers](#)
 - [Powering On a Physical Server](#)
 - [Powering Off a Physical Server](#)
 - [Powering Off Now a Physical Server](#)
 - [Refresh a Physical Server](#)
 - [Restart a Physical Server](#)
 - [Restart Now a Physical Server](#)
 - [Restart a Physical Server to System Setup](#)
 - [Restart Management Controller](#)
 - [Blink the Identify LED](#)
 - [Turn On the Identify LED](#)
 - [Turn Off the Identify LED](#)

Physical Server Management

Management of Physical Servers is available via the following endpoint:

/api/physical_servers

Querying Physical Servers

Querying all physical servers in the system is simply:

GET /api/physical_servers

Getting details on a specific Physical Server:

GET /api/physical_servers/:id

Managing Physical Servers

The following actions are available on Physical Servers:

Action	Description
power_on	Power On
power_off	Power Off
power_off_now	Power Off Now
refresh_now	Refresh
restart	Restart
restart_now	Restart Now
restart_to_sys_setup	Restart to System Setup

Action	Description
restart_mgmt_controller	Restart Management Controller
blink_loc_led	Blink the Identify LED
turn_on_loc_led	Turn On the Identify LED
turn_off_loc_led	Turn Off the Identify LED

Powering On a Physical Server

```
{
  "action" : "power_on"
}
```

Powering Off a Physical Server

```
{
  "action" : "power_off"
}
```

Powering Off Now a Physical Server

```
{
  "action" : "power_off_now"
}
```

Refresh a Physical Server

```
{
  "action" : "refresh"
}
```

Restart a Physical Server

```
{
  "action" : "restart"
}
```

Restart Now a Physical Server

```
{
  "action" : "restart_now"
}
```

Restart a Physical Server to System Setup

```
{
  "action" : "restart_to_sys_setup"
}
```

Restart Management Controller

```
{
  "action" : "restart_mgmt_controller"
}
```

Blink the Identify LED

```
{
  "action" : "blink_loc_led"
}
```

Turn On the Identify LED

```
{  
  "action" : "turn_on_loc_led"  
}
```

Turn Off the Identify LED

```
{  
  "action" : "turn_off_loc_led"  
}
```

- [Picture Management](#)
 - [Creating a Picture](#)
 - [Querying Pictures](#)

Picture Management

Management of Pictures adds support of the following actions:

- [Creating a Picture](#)
- [Querying Pictures](#)

Creating a Picture

POST /api/pictures

```
{  
  "extension": "<extension of image>",  
  "content": "<base 64 encoded image here>"  
}
```

Supported extensions include:

Extension
jpg
png
svg

Querying Pictures

Querying pictures like any other collection:

GET /api/pictures?expand=resources

```
{  
  "name": "pictures",  
  "count": 5,  
  "subcount": 5,  
  "resources": [  
    {  
      "href": "http://localhost:3000/api/pictures/1",  
      "id": "1",  
      "image_href" : "http://localhost:3000/pictures/1.jpg",  
      "extension" : "jpg"  
    },  
    ...  
  ]  
}
```

```

    "href": "http://localhost:3000/api/pictures/4",
    "id": "4",
    "image_href" : "http://localhost:3000/pictures/4.png",
    "extension" : "png",
    "resource_id": "1",
    "resource_type": "ServiceTemplate"
  },
  {
    "href": "http://localhost:3000/api/pictures/5",
    "id": "5",
    "image_href" : "http://localhost:3000/pictures/5.jpg",
    "extension" : "jpg"
  }
],
"actions": [
  {
    "name": "create",
    "method": "post",
    "href": "http://localhost:3000/api/pictures"
  }
]
}

```

This query does not return the pictures themselves, simply metadata about the pictures.

Actually fetching the particular picture is done by getting the image by its image_href.

GET /api/pictures/4

```

{
  "href": "http://localhost:3000/api/pictures/4",
  "id": "4",
  "resource_id": "1",
  "resource_type": "ServiceTemplate",
  "image_href": "http://localhost:3000/pictures/4.jpg"
}

```

In which case the image can simply be downloaded with a GET:

GET http://localhost:3000/pictures/4.jpg

For fetching information about pictures pertaining to services, see the section [Querying Pictures](#) for details.

- [Policy Management](#)
 - [Querying Policies](#)
 - [Querying Policies of a Resource](#)
 - [Creating Policies](#)
 - [Editing Policies](#)
 - [Deleting Policies](#)
 - [Policy and Policy Profiles Actions](#)
 - [Resource Policy Management](#)
 - [Policy specification](#)
 - [Assigning policies to a resource](#)
 - [Response:](#)
 - [Unassigning policies from a resource](#)
 - [Response:](#)
 - [Resolving Policies of a Resource](#)
 - [Response:](#)
 - [Resolving Policy Profiles of a Resource](#)

- [Response:](#)

Policy Management

Management of policies and policy profiles is provided in the following forms:

- Querying of policies via the primary collection:

`/api/policies`

- Querying of policy profiles via the primary collection:

`/api/policy_profiles`

- Assigning and Unassigning policies to resources via the policies subcollection of a resource

`/api/:collection/:id/policies`

and posting **assign** or **unassign** actions to the policies subcollection.

- Assigning and Unassigning policy profiles to resources via the policy_profiles subcollection of a resource

`/api/:collection/:id/policy_profiles`

and posting **assign** or **unassign** actions to the policy_profiles subcollection.

- [Querying Policies](#)
- [Querying Policies of a Resource](#)
- [Creating Policies](#)
- [Editing Policies](#)
- [Deleting Policies](#)
- [Policy and Policy Profiles Actions](#)
- [Resource Policy Management](#)
- [Policy specification](#)
- [Assigning policies to a resource](#)
- [Unassigning policies from a resource](#)
- [Resolving Policies of a Resource](#)
- [Resolving Policy Profiles of a Resource](#)

Querying Policies

Querying all policies in the system is simply:

GET `/api/policies`

```
{
  "name": "policies",
  "count": 20,
```

```

    "subcount": 20,
    "resources": [
      {
        "href": "http://localhost:3000/api/policies/7"
      },
      {
        "href": "http://localhost:3000/api/policies/8"
      },
      ...
    ]
  }
}

```

And expanding the policies, providing additional details:

GET /api/policies?expand=resources

```

{
  "name": "policies",
  "count": 3,
  "subcount": 3,
  "resources": [
    {
      "href": "http://localhost:3000/api/policies/7",
      "id": "7",
      "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "description": "test_control_policy",
      "created_on": "2015-02-04T23:22:32Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    },
    {
      "href": "http://localhost:3000/api/policies/8",
      "id": "8",
      "name": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
      "description": "server_type_control_policy",
      "created_on": "2015-02-05T13:46:39Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    },
    ...
  ]
}

```

Querying Policies of a Resource

To find out which policies are assigned to a resource, the policies subcollection can be expanded for that particular resource as follows:

GET /api/vms/320?expand=policies

This will provide the resources as well as the related policies

```
{
  "href": "http://localhost:3000/api/vms/320",
  "id": "320",
  "vendor": "vmware",
  "name": "aab-vm1",
  "description": "this is a test",
  "raw_power_state": "poweredOn",
  ...
  "policies": [
    {
      "href": "http://localhost:3000/api/vms/320/policies/7",
      "id": "7",
      "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "description": "test_control_policy",
      "created_on": "2015-02-04T23:22:32Z",
      "updated_on": "2015-04-09T13:20:58Z",
      "towhat": "Vm",
      "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
      "created_by": "admin",
      "updated_by": "admin",
      "active": true,
      "mode": "control"
    }
  ]
}
```

One can also simply query the policies of a resource by the subcollection as follows:

GET /api/vms/320/policies

```
{
  "name": "policies",
  "count": 3,
  "subcount": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/320/policies/7"
    }
  ],
  "actions": [
    ...
  ]
}
```

Finding the policies that are part of a policy profile is queried the same as regular resources

GET /api/policy_profiles/:id?expand=policies

or just the policies themselves as follows:

GET /api/policy_profiles/:id/policies

Creating Policies

Policies can be created via a POST to the policies collection or via the **create** action signature which also allows creation of multiple policies in a single request.

POST /api/policies

```
{
  "name" : "sample_policy",
  "description" : "Sample Policy",
  "mode" : "compliance",
  ...
}
```

```

"towhat" : "ManageIQ::Providers::Redhat::InfraManager",
"condition_ids" : [11, 12],
"policy_contents" : [
  {
    "event_id" : 201,
    "actions" : [
      {
        "action_id" : 3001,
        "opts" : { "qualifier" : "failure" }
      }
    ]
  }
]
}

```

or creating multiple policies:

```

{
  "action" : "create",
  "resources" : [
    { "name" : "sample_policy_1", "description" : "Sample Policy 1", ... },
    { "name" : "sample_policy_2", "description" : "Sample Policy 2", ... },
    ...
  ]
}

```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating policies.

Editing Policies

POST /api/policies/:id

```

{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Policy Description"
  }
}

```

or editing multiple policies:

POST /api/policies

```

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/policies/101",
      "description" : "Updated Policy Description 1"
    },
    {
      "href" : "http://localhost:3000/api/policies/102",
      "description" : "Updated Policy Description 2"
    },
    ...
  ]
}

```

Deleting Policies

Policies can be deleted via either the **delete** POST action or via the DELETE HTTP method.

```
POST /api/policies/101
```

```
{  
  "action" : "delete"  
}
```

or simply:

```
DELETE /api/policies/101
```

Deleting multiple policies can be done as follows:

```
POST /api/policies
```

```
{  
  "action" : "delete",  
  "resources" : [  
    { "href" : "http://localhost:3000/api/policies/101" },  
    { "href" : "http://localhost:3000/api/policies/102" },  
    ...  
  ]  
}
```

Policy and Policy Profiles Actions

For managing policies and policy profiles on resource three available actions are available. These are:

Action	Description
assign	Assign a policy or policy profile to the resource
unassign	Unassign a policy or policy profile from the resource
resolve	Resolves a resource policy or policy profile

Resource Policy Management

Policy management on resources can be done by POSTing **assign** and **unassign** actions to the policies or policy_profiles subcollection of resources. Policy management is available on the following primary collections:

Collection
/api/clusters
/api/hosts
/api/policy_profiles
/api/providers
/api/resource_pools
/api/templates
/api/vms

Policy specification

Policies and policy profiles can be specified using one of the following forms:

By href:

```
{  
  "href" : "http://localhost:3000/api/policies/:id"  
}
```

```
{
  "href" : "http://localhost:3000/api/policy_profiles/:id"
}
```

Or by policy Guid:

```
{
  "guid" : "b1fd23bc-acc4-11e4-abc7-b8e85646e742"
}
```

Assigning policies to a resource

Assigning policies or policy_profiles to resources is done by posting an **assign** action against the policies or policy_profiles subcollection of a resource.

POST /api/vms/320/policies

```
{
  "action" : "assign",
  "resource" : { "href" : "http://localhost:3000/api/policies/7" }
}
```

or multiple policies:

```
{
  "action" : "assign",
  "resource" : [
    { "href" : "http://localhost:3000/api/policies/7" },
    { "href" : "http://localhost:3000/api/policies/10" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy: id:'7' description:'test_control_policy'
guid:'b1fd23bc-acc4-11e4-abc7-b8e85646e742'",
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Unassigning policies from a resource

Unassign policies or policy_profiles to resources is done by posting an **unassign** action against the policies or policy_profiles subcollection of a resource.

POST /api/vms/320/policies

```
{
  "action" : "unassign",
  "resource" : { "href" : "http://localhost:3000/api/policies/7" }
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Policy: id:'7' description:'test_control_policy'
guid: 'b1fd23bc-acc4-11e4-abc7-b8e85646e742'",
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Resolving Policies of a Resource

Resolving a resource policy by targeting the subcollection resource directly as follows:

POST /api/vms/320/policies/7

```
{
  "action" : "resolve"
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Resolving Policy: id:'7' description:'test_control_policy'
guid: 'b1fd23bc-acc4-11e4-abc7-b8e85646e742'",
      "result": [
        {
          "id": "7",
          "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "description": "test_control_policy",
          "created_on": "2015-02-04T23:22:32Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [

          ]
        }
      ],
      "href": "http://localhost:3000/api/vms/320",
      "policy_id": "7",
      "policy_href": "http://localhost:3000/api/policies/7"
    }
  ]
}
```

Resolving Policy Profiles of a Resource

Resolving a resource policy profile by targeting the subcollection resource directly as follows:

POST /api/vms/320/policy_profiles/34

```
{
  "action" : "resolve"
}
```

Response:

```
{
  "success": true,
  "message": "Resolving Policy Profile: id:'34' description:'Control Policies'
guid:'f39b25e2-ad3e-11e4-8d4e-b8e85646e742'",
  "result": [
    {
      "id": "34",
      "name": "f39b25e2-ad3e-11e4-8d4e-b8e85646e742",
      "description": "Control Policies",
      "set_type": "MiqPolicySet",
      "created_on": "2015-02-05T13:57:41Z",
      "updated_on": "2015-02-26T13:42:43Z",
      "guid": "f39b25e2-ad3e-11e4-8d4e-b8e85646e742",
      "mode": "control",
      "result": "allow",
      "policies": [
        {
          "id": "7",
          "name": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "description": "test_control_policy",
          "created_on": "2015-02-04T23:22:32Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "b1fd23bc-acc4-11e4-abc7-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [

          ]
        },
        {
          "id": "8",
          "name": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
          "description": "second_test_control_policy",
          "created_on": "2015-02-05T13:46:39Z",
          "updated_on": "2015-04-09T13:20:58Z",
          "towhat": "Vm",
          "guid": "68bfaade-ad3d-11e4-8d4e-b8e85646e742",
          "created_by": "admin",
          "updated_by": "admin",
          "active": true,
          "mode": "control",
          "result": "allow",
          "conditions": [

          ],
          "actions": [

          ]
        }
      ]
    }
  ]
}
```



```

    }
  ]
}
],
"href": "http://localhost:3000/api/vms/320",
"policy_profile_id": "34",
"policy_profile_href": "http://localhost:3000/api/policy_profiles/34"
}

```

- [Policy Collections](#)
- [Policy Actions](#)
 - [Response:](#)
- [Policy Conditions](#)
 - [Response:](#)
- [Policy Events](#)
 - [Response:](#)

Policy Collections

Primary Policy and Policy Profile collections:

`/api/policies`

`/api/policy_profiles`

With Policy Profiles being sets of Policies with the ability to query policies of a specific policy profile via:

`GET /api/policy_profiles/:id?expand=policies`

Or just the subcollection:

`GET /api/policy_profiles/:id/policies`

- [Policy Actions](#)
- [Policy Conditions](#)
- [Policy Events](#)

Policy Actions

Policy actions can be queried as a primary collection or as a subcollection of policies as follows:

`GET /api/policy_actions`

`GET /api/policies/:id?expand=policy_actions`

or simply the subcollection:

`GET /api/policies/:id/policy_actions`

In this example, we're expanding all policy actions

`GET /api/policy_actions?expand=resources`

Response:

```

{
  "name": "policy_actions",

```

```

"count": 33,
"subcount": 33,
"resources": [
  {
    "href": "http://localhost:3000/api/policy_actions/1",
    "id": "1",
    "name": "audit",
    "description": "Generate Audit Event",
    "created_on": "2014-11-20T19:10:05Z",
    "updated_on": "2014-11-20T19:10:05Z",
    "guid": "d619567a-70e8-11e4-a5b4-b8e85646e742",
    "action_type": "default",
    "options": {
    }
  },
  {
    "href": "http://localhost:3000/api/policy_actions/2",
    "id": "2",
    "name": "log",
    "description": "Generate log message",
    "created_on": "2014-11-20T19:10:05Z",
    "updated_on": "2014-11-20T19:10:05Z",
    "guid": "d61c3be2-70e8-11e4-a5b4-b8e85646e742",
    "action_type": "default",
    "options": {
    }
  },
  {
    "href": "http://localhost:3000/api/policy_actions/3",
    "id": "3",
    "name": "evm_event",
    "description": "Show EVM Event on Timeline",
    "created_on": "2014-11-20T19:10:05Z",
    "updated_on": "2014-11-20T19:10:05Z",
    "guid": "d61cb05e-70e8-11e4-a5b4-b8e85646e742",
    "action_type": "default",
    "options": {
    }
  },
  {
    "href": "http://localhost:3000/api/policy_actions/4",
    "id": "4",
    "name": "script",
    "description": "Execute an external script",
    "created_on": "2014-11-20T19:10:05Z",
    "updated_on": "2014-11-20T19:10:05Z",
    "guid": "d61d225a-70e8-11e4-a5b4-b8e85646e742",
    "action_type": "default",
    "options": {
    }
  },
  ...
]
}

```

Policy Conditions

Policy conditions can be queried as a primary collection or as a subcollection of policies as follows:

```
GET /api/conditions
```

```
GET /api/policies/:id?expand=conditions
```

or simply the subcollection:

```
GET /api/policies/:id/conditions
```

In this example, we're expanding the conditions of a policy:

```
GET /api/policies/10?expand=conditions
```

Response:

```
{
  "href": "http://localhost:3000/api/policies/10",
  "id": "10",
  "name": "85dd1f68-ae53-11e4-9c45-b8e85646e742",
  "description": "web_server_policy",
  "created_on": "2015-02-06T22:57:27Z",
  "updated_on": "2015-04-09T13:20:58Z",
  "towhat": "Vm",
  "guid": "85dd1f68-ae53-11e4-9c45-b8e85646e742",
  "created_by": "admin",
  "updated_by": "admin",
  "active": true,
  "mode": "control",
  "conditions": [
    {
      "href": "http://localhost:3000/api/policies/10/conditions/1",
      "id": "1",
      "name": "fe75d0bc-860f-11e4-b12d-b8e85646e742",
      "description": "Memory For Web Server",
      "modifier": "allow",
      "expression": {
        "exp": {
          ">=": {
            "field": "Vm-num_cpu",
            "value": "2"
          }
        },
        "context_type": null
      },
      "towhat": "Vm",
      "created_on": "2014-12-17T17:13:17Z",
      "updated_on": "2015-02-08T15:16:16Z",
      "guid": "fe75d0bc-860f-11e4-b12d-b8e85646e742"
    }
  ]
}
```

Policy Events

Policy events can be queried as a primary collection or as a subcollection of policies as follows:

```
GET /api/events
```

```
GET /api/policies/:id?expand=events
```

or simply the subcollection:

```
GET /api/policies/:id/events
```

In this example, we're expanding all policy events

```
GET /api/events?expand=resources
```

Response:

```
{
  "name": "events",
  "count": 121,
  "subcount": 121,
  "resources": [
    {
      "href": "http://localhost:3000/api/events/1",
      "id": "1",
      "name": "evm_server_start",
      "description": "EVM Server Start",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9edd528-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    {
      "href": "http://localhost:3000/api/events/2",
      "id": "2",
      "name": "evm_server_stop",
      "description": "EVM Server Stop",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9f44c00-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    {
      "href": "http://localhost:3000/api/events/3",
      "id": "3",
      "name": "evm_server_not_responding",
      "description": "EVM Server Not Responding",
      "created_on": "2014-11-20T19:10:11Z",
      "updated_on": "2014-11-20T19:10:11Z",
      "guid": "d9f5dcaa-70e8-11e4-a5b4-b8e85646e742",
      "event_type": "Default"
    },
    ...
  ]
}
```

- [Provider Support](#)
 - [Querying Providers](#)
 - [Cloud Networks](#)
 - [Cloud Subnets](#)
 - [Load Balancers](#)
 - [Virtual Machines](#)
 - [Creating Providers](#)
 - [Specifying Credentials](#)
 - [Specifying Connection Configurations](#)
 - [Editing Providers](#)
 - [Deleting Providers](#)
 - [Refresh Providers](#)
 - [Import Vm to Provider](#)
 - [Foreman Support](#)
 - [Provider Examples](#)

Provider Support

Provider management including CRUD operations as well as the **refresh** action is available via the REST API. Management of providers is through:

`/api/providers`

- [Querying Providers](#)
- [Creating Providers](#)
- [Specifying Credentials](#)
- [Specifying Connection Configurations](#)
- [Editing Providers](#)
- [Deleting Providers](#)
- [Refresh Providers](#)
- [Import Vm to Provider](#)
- [Foreman Support](#)
- [Provider Examples](#)

Querying Providers

Providers can be queried as follows:

`GET /api/providers`

A specific provider can also be queried by its id:

`GET /api/providers/:id`

Foreman type providers can be queried by specifying the `provider_class` as follows:

`GET /api/providers?provider_class=provider`

Cloud Networks

The Cloud Networks of a provider can be queried by asking for the `cloud_networks` subcollection as follows:

`GET /api/providers/:id/cloud_networks`

`GET /api/providers/:id/cloud_networks/:cloud_network_id`

Or expanding the Cloud Networks in addition to the provider as follows:

`GET /api/providers/:id?expand=cloud_networks`

Cloud Subnets

The Cloud Subnets of a provider can be queried by asking for the `cloud_subnets` subcollection as follows:

`GET /api/providers/:id/cloud_subnets`

`GET /api/providers/:id/cloud_subnets/:cloud_subnet_id`

Or expanding the Cloud Subnets in addition to the provider as follows:

```
GET /api/providers/:id?expand=cloud_subnets
```

Load Balancers

Load Balancers of providers can be queried via the `load_balancers` subcollection as follows:

Query all Load Balancers of a provider:

```
GET /api/providers/:id/load_balancers
```

Query a specific Load Balancer of a provider:

```
GET /api/providers/:id/load_balancers/:load_balancer_id
```

Or expanding the Load Balancers for the specific provider as follows:

```
GET /api/providers/:id?expand=load_balancers
```

Virtual Machines

Virtual Machines of providers can be queried via the `vms` subcollection as follows:

Query all VMs of a provider:

```
GET /api/providers/:id/vms
```

Query a specific VM of a provider:

```
GET /api/providers/:id/vms/:vm_id
```

Or expanding the VMs for the specific provider as follows:

```
GET /api/providers/:id?expand=vms
```

Query specific VMs of a provider and get their name and vendor attributes:

```
GET /api/providers/:id/vms?  
filter[]=name='test*&attributes=name,vendor&expand=resources
```

Creating Providers

Creating a provider is done via the **create** action as follows:

```
POST /api/providers  
  
{  
  "action" : "create",  
  "resource" : {  
    # New Provider JSON  
  }  
}
```

And like other **create** actions, the new provider data can simply be posted to the `/api/providers` URL.

```
POST /api/providers  
  
# New Provider JSON
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating providers.

For example, creating a VMware type provider:

POST /api/providers

```
{
  "name" : "vCenter 5",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  "hostname" : "my_vcenter_50",
  "ipaddress" : "192.168.150.1",
  "zone" : { "href" : "http://localhost:3000/api/zone/1" },
  "credentials" : {
    "userid" : "vcenter_admin",
    "password" : "vcenter_password"
  }
}
```

If zone is not specified, the Default zone will be used.

The **type** attribute specifies the supported provider class names which include:

ManageIQ::Providers::Amazon::CloudManager
ManageIQ::Providers::Azure::CloudManager
ManageIQ::Providers::Kubernetes::ContainerManager
ManageIQ::Providers::Microsoft::InfraManager
ManageIQ::Providers::Openshift::ContainerManager
ManageIQ::Providers::Openstack::CloudManager
ManageIQ::Providers::Openstack::InfraManager
ManageIQ::Providers::Lenovo::PhysicalInfraManager
ManageIQ::Providers::Redhat::InfraManager
ManageIQ::Providers::Vmware::CloudManager
ManageIQ::Providers::Vmware::InfraManager

Note:

Please note that the provider **type** classes have changed in the appliance from earlier releases (v2.0.0 of the API). Please refer to the [Provider Types](#) page for a mapping from the old to the new types.

Specifying Credentials

When creating or updating providers, the credentials can be specified as a single default set, or a compound set where additional credentials are necessary, for example AMPQ for OpenStack or Metrics for RHEVM.

Single default credentials set:

```
{
  "name" : "vCenter 50",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  ...
  "credentials" : {
    "userid" : "vc_admin",
    "password" : "vc_password"
  }
}
```

Compound credentials set:

```
{
  "name" : "RHEVM",
  "type" : " ManageIQ::Providers::Redhat::InfraManager",
  ...
  "credentials" : [
    {
      "userid" : "default_userid",
      "password" : "default_password"
    },
    {
      "userid" : "metrics_userid",
      "password" : "metrics_password",
      "auth_type" : "metrics"
    }
  ]
}
```

Specifying Connection Configurations

When creating or updating providers, connection configurations can be set. Connection configurations can be used to specify resources such as an **amqp** event provider for OpenStack, adding Prometheus metrics, or an authentication token.

Prometheus metrics:

```
{
  "name": "Openshift Provider",
  "type": "ManageIQ::Providers::Openshift::ContainerManager",
  ...
  "connection_configurations": [
    {
      "endpoint": {
        "role" : "prometheus",
        "hostname" : "prometheus_host",
        "port" : "443",
        "security_protocol" : "ssl-without-validation",
        "certificate_authority" : "-----BEGIN CERTIFICATE-----",
        "verify_ssl": 0
      },
      "authentication": {
        "role" : "prometheus",
        "auth_key" : "prometheus_auth_key"
      }
    }
  ]
}
```

Amqp event provider:

```
{
  "name": "Openstack Provider",
  "type": "ManageIQ::Providers::Openstack::CloudManager",
  ...
  "connection_configurations": [
    {
      "endpoint": {
        "role" : "amqp",
        "hostname" : "amqphost.com",
        "port" : "5672",
        "security_protocol" : "non-ssl"
      },
    }
  ],
}
```



```

        "authentication": {
            "userid"      : "amqp_userid",
            "password"    : "amqp_password",
            "role"        : "amqp"
        }
    }
]
}

```

Ceilometer event provider:

```

{
    "name"           : "Openstack Provider",
    "type"           : "ManageIQ::Providers::Openstack::CloudManager",
    "hostname"       : "ceilometerhost.com",
    "ipaddress"      : "192.168.150.1",
    "security_protocol" : "ssl",
    ...
    "connection_configurations": {
        "userid"      : "userid",
        "password"    : "password"
    },
    "connection_configurations": [
        "endpoint" : {
            "role"   : "ceilometer"
        }
    ]
}

```

Editing Providers

Editing provider is available via the **edit** action with the **resource** specifying the attributes to update for that provider.

POST /api/providers/:id

```

{
    "action" : "edit" ,
    "resource" : {
        "hostname" : "new_vcenter_50",
        "ipaddress" : "192.168.150.2"
    }
}

```

Note:

type is a restricted attribute and cannot be edited.

Deleting Providers

Deleting a provider is done via the delete method:

DELETE /api/providers/:id

or via the **delete** action as follows:

POST /api/providers/:id

```

{
    "action" : "delete"
}

```

or by href:

POST /api/providers

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/:id" }
  ]
}
```

Refresh Providers

Performing a refresh of providers is done via the **refresh** action.

POST /api/providers/:id

```
{
  "action" : "refresh"
}
```

You can also refresh multiple providers:

POST /api/providers

```
{
  "action" : "refresh",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/:id" },
    { "href" : "http://localhost:3000/api/providers/:id" }
  ]
}
```

Import Vm to Provider

Importing a VM into a provider is supported via the *import_vm* action.

This is currently supported for importing a Vm from a VMware vCenter provider into a Red Hat Virtualization provider.

POST /api/providers/:id

```
{
  "action" : "import_vm"
  "resource" : {
    "source" : { "href" : "http://localhost:3000/api/vms/11" },
    "target" : {
      "name" : "new_vm_name",
      "cluster" : { "href" : "http://localhost:3000/api/clusters/201" },
      "data_store" : { "href" : "http://localhost:3000/api/data_stores/301" },
      "sparse" : true
    }
  }
}
```

Foreman Support

Management of Foreman providers is provided as a separate class of providers. This is still accessed via the **/api/providers** entry point, but enabled by a new parameter:

GET|POST|DELETE /api/providers?provider_class=provider

The **provider_class** parameter is supported with **/api/provider** for all CRUD operations including the **refresh** action.

An example of creating a Foreman provider:

POST /api/providers?provider_class=provider

```
{
  "type"      : "ManageIQ::Providers::Foreman::Provider",
  "name"      : "new_foreman_111",
  "url"       : "100.200.300.111",
  "credentials" : {
    "userid"   : "foreman_admin",
    "password" : "foreman_password"
  }
}
```

When creating providers with provider_class as *provider*, the **type** attribute specifies the supported provider class names which include:

ManageIQ::Providers::AnsibleTower::Provider
ManageIQ::Providers::Foreman::Provider
ManageIQ::Providers::Openstack::Provider

Note:

When managing providers of *provider* provider_class, Tagging and Policy management does not currently apply. So requests including the tags, policies and policy_profiles subcollections will be rejected.

Provider Examples

Below you will find example requests for creating various providers.

Ansible Tower Automation Manager:

```
{
  "type": "ManageIQ::Providers::AnsibleTower::AutomationManager",
  "name": "API test",
  "api_version": "3.1.2",
  "provider_id": 10000000000012,
  "credentials": {
    "userid": "userid",
    "password": "password"
  }
}
```

AWS:

```
{
  "type": "ManageIQ::Providers::Amazon::CloudManager",
  "provider_region": "us-east-1",
  "name": "Amazon Example",
  "credentials": {
    "userid": "<access key id>",
    "password": "<secret access key>"
  }
}
```

Azure:

```
{
  "type": "ManageIQ::Providers::Azure::CloudManager",
  "name": "Azure Example",
  "provider_region": "eastus",
  "uid_ems": "<tenant_id>",
  "subscription": "<subscription_id>",
  "credentials": {
    "userid": "<client id>",
    "password": "<client key>"
  }
}
```

Kubernetes Container Manager:

```
{
  "type": "ManageIQ::Providers::Kubernetes::ContainerManager",
  "name": "Kubernetes Test",
  "connection_configurations": [
    {
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "6443",
        "security_protocol": "ssl"
      },
      "authentication": {
        "auth_type": "bearer",
        "auth_key": "<token_here>"
      }
    }
  ]
}
```

Microsoft System Center VMM (SCVMM):

```
{
  "type": "ManageIQ::Providers::Microsoft::InfraManager",
  "name": "SCVMM API example",
  "api_version": "2.1.0",
  "connection_configurations": [
    {
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "443",
        "security_protocol": "ssl"
      },
      "authentication": {
        "username": "username",
        "password": "password"
      }
    }
  ]
}
```

OpenShift Container Manager:

```
{
  "name": "ocp-example",
  "port": 8443,
  "type": "ManageIQ::Providers::Openshift::ContainerManager",
  "hostname": "ocp.example.com",
  "connection_configurations": [
    {
      "endpoint": {
        "role": "default",
```

```

        "security_protocol": "ssl-without-validation",
        "verify_ssl": 0
    },
    "authentication": {
        "authtype": "bearer",
        "type": "AuthToken",
        "auth_key": "<auth key>"
    }
}
]
}

```

OpenStack Cloud Manager:

```

{
  "type": "ManageIQ::Providers::Openstack::CloudManager",
  "name": "API Example",
  "api_version": "v2",
  "connection_configurations": [
    {
      "endpoint": {
        "role": "default",
        "hostname": "00.0.00.001",
        "port": "5000",
        "security_protocol": "non-ssl"
      },
      "authentication": {
        "type": "ManageIQ::Providers::Openstack::CloudManager::AuthKeyPair",
        "fingerprint": "fingerprint"
      }
    }
  ]
}

```

OpenStack Infra Manager:

```

{
  "type": "ManageIQ::Providers::Openstack::InfraManager",
  "name": "OSP API example",
  "api_version": "v2",
  "connection_configurations": [
    {
      "endpoint": {
        "hostname": "00.0.00.00",
        "port": "5000",
        "security_protocol": "non-ssl"
      },
      "authentications": {
        "username": "username",
        "password": "password"
      }
    }
  ]
}

```

Red Hat Virtualization Manager:

```

{
  "type": "ManageIQ::Providers::Redhat::InfraManager",
  "name": "Sample Rhev",
  "port": 5000,
  "hostname": "sample_rhev.provider.com",
  "ipaddress": "100.200.300.4",
  "security_protocol": "kerberos",

```

```

"certificate_authority": "-----BEGIN CERTIFICATE-----...",
"credentials": {
  "userid": "uname",
  "password": "password"
}
}

```

VMware Infra Manager:

```

{
  "type": "ManageIQ::Providers::Vmware::InfraManager",
  "name": "Sample VMWare provider",
  "hostname": "sample_vmware.provider.com",
  "ipaddress": "100.200.300.1",
  "credentials": {
    "userid": "uname",
    "password": "pword"
  }
}

```

VMware Cloud Manager:

```

{
  "type": "ManageIQ::Providers::Vmware::CloudManager",
  "api_version": "9.0",
  "name": "Api Test",
  "connection_configurations": [
    {
      "endpoint": {
        "port": "443",
        "hostname": "00.0.00.000"
      },
      "authentication": {
        "username": "username",
        "password": "pword"
      }
    }
  ]
}

```

- [Provision Requests](#)
 - [Creating Requests](#)
 - [Provision Requests Attribute Groups](#)
 - [Editing Requests](#)
 - [Response:](#)
 - [Approving or Denying Requests](#)
 - [Approving a Request](#)
 - [Denying a Request](#)
 - [Monitoring Requests](#)
 - [Response:](#)
 - [Editing Request Tasks](#)

Provision Requests

Provisioning requests are made available via the following endpoint.

`/api/provision_requests`

- [Creating Requests](#)

- [Editing Requests](#)
- [Approving or Denying Requests](#)
- [Monitoring Requests](#)
- [Editing Request Tasks](#)

Creating Requests

Either by specifying a *create* action or by posting the request directly to **/api/provision_requests**.

As an example:

```
{
  "version" : "1.1",
  "template_fields" : {
    "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
  },
  "vm_fields" : {
    "number_of_cpus" : 1,
    "vm_name" : "aab_test_vm1",
    "vm_memory" : "1024",
    "vlan" : "nic1"
  },
  "requester" : {
    "user_name" : "jdoe",
    "owner_first_name" : "John",
    "owner_last_name" : "Doe",
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  },
  "tags" : {
    "network_location" : "Internal",
    "cc" : "001"
  },
  "additional_values" : {
    "request_id" : "1001"
  },
  "ems_custom_attributes" : { },
  "miq_custom_attributes" : { }
}
```

Provision Requests Attribute Groups

Attribute Group	Type	Description
version	string	Interface version. Defaults to 1.1
template_fields	hash	Fields used to find template virtual machine. Provide any or all fields. Supply a guid or ems_guid to protect against matching same-named templates on different Providers within the appliance. Supported fields are: name=[VM Template Name] Example: template_1 guid=[guid value from vms resource] ems_guid=[uid_ems value from vms resource]
vm_fields	hash	Allows for the setting of properties on the Catalog, Hardware, Network, Customize, and Schedule tabs in the Provisioning dialog
requester	hash	Allows for the setting of properties on the Requester tab in the Provisioning dialog

Attribute Group	Type	Description
tags	hash	Tags to apply to newly created VM Example: network_location=Internal cc=001
additional_values	hash	Additional values are name-value pairs stored with a provision request, but not used by the core provisioning code. These values are usually referenced from automate methods for custom processing Example: Store a request_id from an external system so the system can be notified during the provisioning process
ems_custom_attributes	hash	Custom attributes applied to the virtual machine through the Provider as part of provisioning
miq_custom_attributes	hash	Custom attributes applied to the virtual machine and stored in the database as part of provisioning

Editing Requests

Updating provision requests can be performed by posting an *edit* action as follows:

POST /api/provision_requests/11

```
{
  "action" : "edit",
  "resource" : {
    "options" : {
      "custom_var1" : "custom value1"
    }
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/provision_requests/11",
  "id": "11",
  "description": "Provision from [template1] to [###]",
  "approval_state": "pending_approval",
  "type": "MiqProvisionRequest",
  "created_on": "2015-04-14T17:36:30Z",
  "updated_on": "2015-04-14T18:32:30Z",
  "requester_id": "88913",
  "requester_name": "API User",
  "request_type": "template",
  "request_state": "pending",
  "message": "VM Provisioning - Request Created",
  "status": "Ok"
  "options": {
    "use_pre_dialog": false,
    "request_type": "template",
    "miq_request_dialog_name": "miq_provision_dialogs",
    "src_vm_id": [
      109996,
      "template1"
    ],
    "src_vm_nics": [],
    "src_vm_lans": [],
    "src_ems_id": [
      59136,
      "ems_00000000000002"
    ]
  },
}
```



```

    "placement_auto": [
      true,
      1
    ],
    "vm_tags": [],
    "ws_values": {
    },
    "ws_ems_custom_attributes": {
    },
    "ws_miq_custom_attributes": {
    },
    "tags": {
    },
    "userid": "admin",
    "source_id": "109996",
    "source_type": "VmOrTemplate",
    "custom_var1" : "custom value1"
  }
}

```

Approving or Denying Requests

If a provision is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the provision request resource.

Approving a Request

POST /api/provision_requests/:id

```

{
  "action" : "approve",
  "reason" : "reason for approval"
}

```

Denying a Request

POST /api/provision_requests/:id

```

{
  "action" : "deny",
  "reason" : "reason for denial"
}

```

Monitoring Requests

Once a provision request is created, the response result will include the queryable provision request itself, i.e. /api/provision_requests/:id

Response:

```

{
  "results": [
    {
      "id": "3068",
      "description": "Provision from [template1] to [###]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-04-14T17:36:30Z",
      "updated_on": "2015-04-14T17:36:30Z",
      "requester_id": "88913",
      "requester_name": "API User",
    }
  ]
}

```

```

    "request_type": "template",
    "request_state": "pending",
    "message": "VM Provisioning - Request Created",
    "status": "Ok"
    "options": {
      "use_pre_dialog": false,
      "request_type": "template",
      "miq_request_dialog_name": "miq_provision_dialogs",
      "src_vm_id": [
        109996,
        "template1"
      ],
      "src_vm_nics": [],
      "src_vm_lans": [],
      "src_ems_id": [
        59136,
        "ems_00000000000002"
      ],
      "placement_auto": [
        true,
        1
      ],
      "vm_tags": [],
      "ws_values": {
      },
      "ws_ems_custom_attributes": {
      },
      "ws_miq_custom_attributes": {
      },
      "tags": {
      }
    },
    "userid": "admin",
    "source_id": "109996",
    "source_type": "VmOrTemplate"
  }
]
}

```

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state. Specifically,

```
GET /api/provision_requests/3068
```

Note:

The requests tasks of a provisioning request can also be queried by expanding the request_tasks subcollection as follows:

```
GET /api/provision_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/provision_requests/:id/request_tasks?expand=resources
```

Editing Request Tasks

Request tasks for Provision Requests can be updated as follows:

```

POST /api/provision_requests/:id/request_tasks/:request_task_id
{
  "action" : "edit",

```

```

"resource" : {
  "options" : {
    "request_param_a" : "value_a",
    "request_param_b" : "value_b"
  }
}
}

```

For a list of attributes available for the provision request attribute groups please refer to the [Provision Request Supported Attributes](#) in the appendices section.

- [Request Support](#)
 - [Creating Requests](#)
 - [Querying Requests](#)
 - [Editing Requests](#)
 - [Approving or Denying Requests](#)
 - [Approving a Request](#)
 - [Denying a Request](#)
 - [Request Types](#)

Request Support

Support for Requests in the API is provided and implemented by specific request types as listed here:

API Collection	Request Type
/api/automation_requests	Automation Requests
/api/provision_requests	Provision Requests
/api/service_requests	Service Requests

A lower level support for creating and updating requests is also provided by the following:

/api/requests

- [Creating Requests](#)
- [Querying Requests](#)
- [Editing Requests](#)
- [Approving or Denying Requests](#)
- [Request Types](#)

Creating Requests

Either by specifying a *create* action or by posting the request directly to **/api/requests**.

As an example:

POST /api/requests

```

{
  "options" : {
    "request_type" : "service_reconfigure",
    "src_id" : "4",
    "service_param1" : "updated_value1",
    "service_param2" : "updated_value2"
  },

```

```

"requester" : {
  "user_name" : "admin"
},
"auto_approve" : true | false
}

```

Response:

```

{
  "results": [
    {
      "id": "97",
      "description": "Service Reconfigure for: svc4",
      "approval_state": "pending_approval",
      "type": "ServiceReconfigureRequest",
      "created_on": "2016-11-10T21:39:08Z",
      "updated_on": "2016-11-10T21:39:08Z",
      "requester_id": "1",
      "requester_name": "Administrator",
      "request_type": "service_reconfigure",
      "request_state": "pending",
      "message": "Service Reconfigure - Request Created",
      "status": "Ok",
      "options": {
        "request_type": "service_reconfigure",
        "src_id": "4",
        "service_param1": "updated_value1",
        "service_param2": "updated_value2"
      },
      "userid": "admin",
      "source_id": "4",
      "source_type": "Service",
      "tenant_id": "1",
      "process": true
    }
  ]
}

```

Querying Requests

Once a request is created, the response result as shown above will include the id of request itself.

In the above example, the request could be queried periodically until the **request_state** reaches the **finished** state.

GET /api/requests/97

NOTE: The requests tasks of a request can also be queried by expanding the request_tasks subcollection:

GET /api/requests/:id?expand=request_tasks

Editing Requests

Updating requests is provided by specifying an *edit* action as follows:

POST /api/requests/97

```

{
  "action" : "edit",
  "resource" : {
    "options" : {
      "service_param1" : "new_updated_value1",
      "service_param2" : "new_updated_value2"
    }
  }
}

```

```

    },
    "requester" : {
        "user_name" : "admin"
    }
}
}

```

Response:

```

{
  "href": "http://localhost:3000/api/requests/97",
  "id": "97",
  "options": {
    "request_type": "service_reconfigure",
    "src_id": "4",
    "service_param1": "new_updated_value1",
    "service_param2": "new_updated_value2"
  },
  "description": "Service Reconfigure for: svc3",
  "approval_state": "pending_approval",
  "type": "ServiceReconfigureRequest",
  "created_on": "2016-11-10T21:39:08Z",
  "updated_on": "2016-11-10T22:11:03Z",
  "requester_id": "1",
  "requester_name": "Administrator",
  "request_type": "service_reconfigure",
  "request_state": "pending",
  "message": "Service Reconfigure - Request Created",
  "status": "Ok",
  "userid": "admin",
  "source_id": "4",
  "source_type": "Service",
  "tenant_id": "1",
  "process": true
}

```

Approving or Denying Requests

If a request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the request resource.

Approving a Request

POST /api/requests/:id

```

{
  "action" : "approve",
  "reason" : "reason for approval"
}

```

Denying a Request

POST /api/requests/:id

```

{
  "action" : "deny",
  "reason" : "reason for denial"
}

```

Request Types

The details for creating each type of request using the lower level API, including validation of *request_type* and necessary options is driven by the model.

The basic structure for creating the request is as follows:

```
{
  "options" : {
    "request_type" : "...",
    ...
  },
  "requester" : {
    "user_name" : "..."
  },
  "auto_approve" : true | false
}
```

The list of supported *request_type* values as of this writing is shown here below:

request_type	Description
automation	Automation
clone_to_service	Service Provision
clone_to_template	VM Publish
clone_to_vm	VM Clone
host_pxe_install	Host Provision
provision_via_foreman	Foreman Provision
service_reconfigure	Service Reconfigure
template	VM Provision
vm_migrate	VM Migrate
vm_reconfigure	VM Reconfigure

- [Reports Support](#)
 - [Querying Reports](#)
 - [Running Reports](#)
 - [Request:](#)
 - [Response:](#)
 - [Scheduling Reports](#)
 - [Example: "One off" report](#)
 - [Request](#)
 - [Response](#)
 - [Requesting Report Result Downloads](#)
 - [Request:](#)
 - [Response:](#)
 - [Importing Reports](#)
 - [Request:](#)
 - [Response:](#)

Reports Support

Support for querying, importing and running Reports is now available. Access to reports and report results is through the following collections:

/api/reports

`/api/results`

- [Querying Reports](#)
- [Running Reports](#)
- [Scheduling Reports](#)
- [Requesting Report Result Downloads](#)
- [Importing Reports](#)

Querying Reports

Querying Available Reports

GET `/api/reports`

Querying a Single Report

GET `/api/reports/:id`

Querying Results of a Particular Reports

GET `/api/reports/:id/results`

Optionally, expanding all results

GET `/api/reports/:id/results?expand=resources`

Querying a single result of a report

GET `/api/reports/:id/results/:result_id`

Querying report results

GET `/api/results`

Running Reports

Running reports is done via the **run** action on a report resource as follows:

POST `/api/reports/:id`

```
{
  "action" : "run"
}
```

Example for running a report on the following:

```
{
  "href": "http://localhost:3000/api/reports/80",
  "id": "80",
  "name": "VM Relationships"
}
```

Request:

POST `/api/reports/80`

```
{
  "action" : "run"
}
```

Response:

```
{
  "success": true,
  "message": "running report 80",
  "href": "http://localhost:3000/api/reports/80",
  "result_id": "27",
  "result_href": "http://localhost:3000/api/results/27",
  "task_id": "726",
  "task_href": "http://localhost:3000/api/tasks/726"
}
```

The report run's task can be queried periodically via the task_href specified above as follows:

GET /api/tasks/726

```
{
  "href": "http://localhost:3000/api/tasks/726",
  "id": "726",
  "name": "Generate Report: 'VM Relationships'",
  "state": "Queued",
  "status": "Ok",
  "message": "Task has been queued",
  "userid": "system",
  "created_on": "2015-11-09T18:44:41Z",
  "updated_on": "2015-11-09T18:44:41Z"
}
```

```
{
  "href": "http://localhost:3000/api/tasks/726",
  "id": "726",
  "name": "Generate Report: 'VM Relationships'",
  "state": "Finished",
  "status": "Ok",
  "message": "Generating report complete",
  "userid": "system",
  "created_on": "2015-11-09T18:44:41Z",
  "updated_on": "2015-11-09T18:47:13Z"
}
```

With the results queried via the result_href provided above:

GET /api/results/27

```
{
  "href": "http://localhost:3000/api/results/27",
  "id": "27",
  "name": "VM Relationships",
  "miq_report_id": "80",
  "miq_task_id": "726",
  "userid": "system",
  "report_source": "Requested by user",
  "db": "Vm",
  "report": {
    "name": "VM Relationships",
    "title": "VM Relationships",
    "rpt_group": "Relationships - Virtual Machines, Folders, Clusters",
    "rpt_type": "Default",
    "priority": 10,
    "db": "Vm",
    "cols": [
      "name",
      "v_owning_datacenter",
      "v_owning_folder",

```



```

    "v_owning_resource_pool"
  ],
  "include": {
    "ext_management_system": {
      "columns": [
        "name"
      ]
    },
    "host": {
      "columns": [
        "v_owning_cluster",
        "hostname"
      ]
    },
    "storage": {
      "columns": [
        "name"
      ]
    }
  },
  "col_order": [
    "name",
    "ext_management_system.name",
    "v_owning_datacenter",
    "host.v_owning_cluster",
    "host.hostname",
    "v_owning_folder",
    "v_owning_resource_pool",
    "storage.name"
  ],
  "headers": [
    "VM Name",
    "Provider Name",
    "Datacenter",
    "Cluster",
    "Host",
    "Folder",
    "Resource Pool",
    "Datastore"
  ],
  "conditions": {
    "exp": {
      "IS NOT EMPTY": {
        "field": "Vm.host-name"
      }
    },
    "col_details": {
      "Vm.host-name": {
        "data_type": "string",
        "virtual_reflection": false,
        "virtual_column": false,
        "excluded_by_preprocess_options": false,
        "tag": false,
        "include": {
          "host": {
            "v_owning_cluster": true,
            "hostname": true,
            "v_owning_folder": true,
            "v_owning_resource_pool": true,
            "storage.name": true
          }
        }
      },
      "format_sub_type": "string"
    }
  },
  "pexp": {
    "IS NOT EMPTY": {
      "field": "Vm.host-name"
    }
  }
}

```

```

    },
    "ruby": "<value ref=vm, type=string>/virtual/host/name</value> != \"\"\"",
  },
  "order": "Ascending",
  "sortby": [
    "name"
  ],
  "filename": "450_Relationships - Virtual Machines, Folders, Clusters/010_VMs
Relationships.yaml",
  "file_mtime": "2015-08-19T13:23:50Z",
  "template_type": "report",
  "col_formats": [

  ],
  "rpt_options": {
  },
  "extras": {
    "total_html_rows": 1709
  },
  "user_categories": [

  ],
  "trend_data": {
  }
},
"created_on": "2015-11-09T18:52:48Z",
"last_run_on": "2015-11-09T18:52:56Z",
"last_accessed_on": "2015-11-09T18:52:56Z",
"result_set": [
  {
    "name": "aab-vm1",
    "v_owning_datacenter": "",
    "v_owning_folder": "",
    "v_owning_resource_pool": "",
    "host.v_owning_cluster": "",
    "host.hostname": "test-host.testdomain.com",
    "storage.name": "StarStorage1",
    "id": "320"
  },
  {
    "name": "aab_demo_vm",
    "v_owning_datacenter": "",
    "v_owning_folder": "",
    "v_owning_resource_pool": "",
    "host.v_owning_cluster": "",
    "host.hostname": "test-host2.testdomain.com",
    "storage.name": "StarStorage2",
    "id": "1"
  },
  ...
]
}

```

Note that one can query all the results run for a report as follows:

```
GET /api/reports/80/results?expand=resources&attributes=created_on
```

```

{
  "name": "results",
  "count": 27,
  "subcount": 2,
  "resources": [
    {
      "href": "http://localhost:3000/api/reports/80/results/27",

```

```

    "id": "27",
    "created_on": "2015-11-09T18:52:48Z"
  },
  {
    "href": "http://localhost:3000/api/reports/80/results/13",
    "id": "13",
    "created_on": "2015-07-29T18:24:35Z"
  }
]
}

```

Scheduling Reports

Scheduling a report is done via the **schedule** action on a report resource as follows

POST /api/reports/:id

```

{
  "action" : "schedule"
  "resource": {
    "send_email": true|false,
    "start_date": DateTime,
    "time_zone": TZ,
    "interval": {
      "unit": "minutely"|"hourly"|"daily"|"weekly"|"monthly"|"once",
      "value": "once"
    },
    ...
  }
}

```

Example: “One off” report

Request

POST /api/reports/5

```

{
  "action": "schedule",
  "resource": {
    "name": "Chargeback Report Schedule - 2000/01/01",
    "description": "Chargeback Report 2000 New Year Report",
    "resource_type": "MiqReport",
    "start_date": "2000-01-01T00:00:00Z",
    "time_zone": "UTC",
    "interval": { "unit": "once" },
    "enabled": true
  }
}

```

Response

```

{
  "success": true,
  "message": "scheduling of report 5",
  "schedule_id": "10",
  "schedule_href": "http://localhost:3000/api/reports/5/schedules/10",
  "href": "http://localhost:3000/api/reports/5"
}

```

Requesting Report Result Downloads

Requesting a **txt**, **csv** or **pdf** download of the report results can be done via the **request_download** action requiring the desired *result_type* as follows:

Request:

POST /api/results/22

```
{
  "action" : "request_download",
  "result_type" : "csv"
}
```

This triggers an async request of the report and returns an action response as follows:

Response:

```
{
  "success": true,
  "message": "Requesting a download of a csv report for Result id:22 name:VMware VM Summary",
  "task_id": "46",
  "task_href": "http://localhost:3000/api/tasks/46",
  "task_results_href": "http://localhost:3000/api/tasks/46/task_results"
}
```

The task can simply be queried for a *Finished* state and the results then fetched via the *task_results_href* provided above.

Querying for the task completion:

GET /api/tasks/46

```
{
  "href": "http://localhost:3000/api/tasks/46",
  "id": "46",
  "name": "Generate Report result [csv]: 'VMware VM Summary'",
  "state": "Finished",
  "status": "Ok",
  "message": "Generate Report result [csv]",
  "userid": "admin",
  "created_on": "2020-05-06T17:15:29Z",
  "updated_on": "2020-05-06T17:15:30Z",
  "pct_complete": null,
  "context_data": {
    "result_id": "22",
    "result_type": "csv",
    "session_id": "1cae5e9c-9b3e-4ee1-a286-bc2a0131233d-7f4bc905"
  },
  "results": null,
  "miq_server_id": "3",
  "identifier": null,
  "started_on": "2020-05-06T17:15:30Z",
  "zone": null,
  "actions": [
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/tasks/46"
    },
    {
      "name": "delete",
      "method": "delete",

```

```

    "href": "http://localhost:3000/api/tasks/46"
  }
]
}

```

And the results can be fetched as follows:

GET /api/tasks/46/task_results

This will return the report with the appropriately typed **Content-Type**, *application/txt*, *application/csv*, or *application/pdf* as per the requested download.

With a **request_download** action targeted at the collection level, one can also request the download for multiple report results as follows:

POST /api/results

```

{
  "action" : "request_download",
  "resources" : [
    { "id" : "15", "result_type" : "txt" },
    { "id" : "17", "result_type" : "txt" },
    { "id" : "22", "result_type" : "csv" }
  ]
}

{
  "results" : [
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:15",
      "name": "VMware VM Summary",
      "task_id": "54",
      "task_href": "http://localhost:3000/api/tasks/54",
      "task_results_href": "http://localhost:3000/api/tasks/54/task_results"
    },
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:17",
      "name": "VMware VM Summary",
      "task_id": "55",
      "task_href": "http://localhost:3000/api/tasks/55",
      "task_results_href": "http://localhost:3000/api/tasks/55/task_results"
    },
    {
      "success": true,
      "message": "Requesting a download of a txt report for Result id:22",
      "name": "VMware VM Summary",
      "task_id": "56",
      "task_href": "http://localhost:3000/api/tasks/56",
      "task_results_href": "http://localhost:3000/api/tasks/56/task_results"
    }
  ]
}

```

Each can then be individually queried and downloaded upon completion.

Importing Reports

Importing reports to the system can be done by posting an **import** action to the reports collection as follows:

POST /api/reports

```
{
  "action" : "import",
  "resource" : {
    "report" : { JSON serialized report },
    "options" : { "save" : true }
  }
}
```

Here's an example:

Request:

POST /api/reports

```
{
  "action" : "import",
  "resource" : {
    "report" : {
      "menu_name" : "Test Report",
      "col_order" : ["col1", "col2", "col3"],
      "cols" : ["col1", "col2", "col3"],
      "rpt_type" : "Custom",
      "title" : "Test Report",
      "db" : "My::Db",
      "rpt_group" : "Custom"
    },
    "options" : { "save" : true }
  }
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Imported Report: [Test Report]",
      "result": {
        "id": "199",
        "name": "Test Report",
        "title": "Test Report",
        "rpt_group": "Custom",
        "rpt_type": "Custom",
        "db": "My::Db",
        "cols": [
          "col1",
          "col2",
          "col3"
        ],
        "col_order": [
          "col1",
          "col2",
          "col3"
        ],
        "created_on": "2015-11-09T19:44:37Z",
        "updated_on": "2015-11-09T19:44:37Z",
        "miq_group_id": "1",
        "user_id": "1"
      }
    ]
  }
}
```

- [Role Management](#)
 - [Querying Roles](#)
 - [Creating Roles](#)
 - [Editing Roles](#)
 - [Assigning features to a role](#)
 - [Unassigning features from a role](#)
 - [Deleting Roles](#)

Role Management

Management of User Roles is provided via the following collection:

`/api/roles`

as well as the associated product features via:

`/api/features`

Full CRUD actions on roles is available:

- [Querying Roles](#)
- [Creating Roles](#)
- [Editing Roles](#)
- [Deleting Roles](#)

Querying Roles

Querying all roles in the system is simply:

`GET /api/roles`

Getting details on a specific role:

`GET /api/roles/:id`

Including the features it entitles:

`GET /api/roles/:id?expand=features`

Or simply querying the features entitled by the role:

`GET /api/roles/:id/features?expand=resources`

Querying all product features of the system that can be assigned to a role.

`GET /api/features`

Creating Roles

Roles can be created via a POST to the roles collection or via the **create** action signature which also allows creation of multiple roles in a single request.

`POST /api/roles`

```
{
  "action" : "create",
```

```

"resource" : {
  "name" : "sample_role",
  "settings" : { "restrictions" : { "vms" : "user" } },
  "features" : [
    { "identifier" : "vm_explorer" },
    { "identifier" : "ems_infra_tag" },
    { "identifier" : "miq_report_run" }
  ]
}
}

```

Note:

restrictions for **vms** can be either *user* or *user_or_group*

features can be specified via **identifier**, **href**, or **id**

or creating multiple roles:

```

{
  "action" : "create",
  "resources" : [
    { "name" : "sample_role1", ... },
    { "name" : "sample_role2", ... },
    ...
  ]
}

```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Roles.

Editing Roles

POST /api/roles/:id

```

{
  "action" : "edit",
  "resource" : {
    "name" : "updated_sample_role",
    "settings" : { "restrictions" : { "vms" : "user_or_group" } }
  }
}

```

or editing multiple roles:

POST /api/roles

```

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/roles/101",
      "name" : "updated_sample_role1"
    },
    {
      "href" : "http://localhost:3000/api/roles/102",
      "name" : "updated_sample_role2"
    },
    ...
  ]
}

```


Assigning features to a role

POST /api/roles/:id/features

```
{
  "action" : "assign",
  "resource" : {
    "identifier" : "miq_request_view"
  }
}
```

or assigning multiple features:

```
{
  "action" : "assign",
  "resources" : [
    { "identifier" : "miq_request_view" },
    { "identifier" : "storage_manager_show_list" },
    ...
  ]
}
```

Unassigning features from a role

POST /api/roles/:id/features

```
{
  "action" : "unassign",
  "resource" : {
    { "identifier" : "miq_request_view" }
  }
}
```

or unassigning multiple features:

```
{
  "action" : "unassign",
  "resources" : [
    { "identifier" : "miq_request_view" },
    { "identifier" : "storage_manager_show_list" },
    ...
  ]
}
```

Deleting Roles

Non system roles (i.e. read_only false) can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/roles/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/roles/101

Deleting multiple roles can be done as follows:

POST /api/roles

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/roles/101" },
    { "href" : "http://localhost:3000/api/roles/102" },
    ...
  ]
}
```

- [Security Groups](#)
 - [Querying Security Groups](#)
 - [Managing Security Groups](#)
 - [Add Security Group](#)
 - [Remove Security Group](#)

Security Groups

Management of Security Groups is available via the following collection:

/api/security_groups

Querying Security Groups

Querying all available Security Groups:

GET /api/security_groups

Getting details on a specific Security Group:

GET /api/security_groups/:id

Querying the Security Groups of a specific resource is also possible for the following collections:

Collection
/api/cloud_tenants
/api/instances
/api/providers
/api/vms

GET /api/cloud_tenants/:id/security_groups

GET /api/cloud_tenants/:id/security_groups/:security_group_id

GET /api/instances/:id/security_groups

GET /api/instances/:id/security_groups/:security_group_id

GET /api/providers/:id/security_groups

GET /api/providers/:id/security_groups/:security_group_id

GET /api/vms/:id/security_groups

GET /api/vms/:id/security_groups/:security_group_id

Managing Security Groups

Management of Security Groups is available for Instances with the following actions:

Action	Description
add	Add security group(s) to the Instance

Action	Description
remove	Remove security group(s) from the Instance

Add Security Group

Security Groups can be added to Instances individually:

POST /api/instances/:id/security_groups

```
{
  "action" : "add",
  "resource" : {
    "name" : "security_group_a"
  }
}
```

Or in bulk as follows:

```
{
  "action" : "add",
  "resources" : [
    { "name" : "security_group_a" },
    { "name" : "security_group_b" }
  ]
}
```

Remove Security Group

Security groups can be removed from Instances individually:

POST /api/instances/:id/security_groups

```
{
  "action" : "remove",
  "resource" : {
    "name" : "security_group_c"
  }
}
```

Or in bulk as follows:

```
{
  "action" : "remove",
  "resources" : [
    { "name" : "security_group_d" },
    { "name" : "security_group_e" },
    { "name" : "security_group_f" }
  ]
}
```

- [Service Management](#)
 - [Service Catalogs](#)
 - [Creating Service Catalogs](#)
 - [Editing Service Catalogs](#)
 - [Assigning Service Templates](#)
 - [Ordering Services Service Templates](#)
 - [Refresh Service Dialog Fields](#)
 - [Refresh Dialog Fields of Service Templates Service Dialogs](#)
 - [Refresh Dialog Fields of Service Dialogs](#)
 - [Deleting Service Catalogs](#)

- [Service Templates](#)
 - [Creating Service Templates](#)
 - [Editing Service Templates](#)
 - [Deleting Service Templates](#)
- [Services ===](#)
 - [Creating Services =====](#)
 - [Service Actions =====](#)
 - [Service Editing](#)
 - [Adding and Removing Resources from Services](#)
 - [Adding Provider VMs to Services](#)
 - [Requesting a Chargeback Report for a Service](#)
 - [Service Retiring \(now\)](#)
 - [Service Retiring \(future\)](#)
 - [Service Reconfiguring](#)
 - [Starting Services](#)
 - [Stopping Services](#)
 - [Suspending Services](#)
 - [Deleting Services](#)
- [Service Requests ===](#)
 - [Querying Service Requests](#)
 - [Approving or Denying Service Requests](#)
 - [Approving a Service Request](#)
 - [Denying a Service Request](#)
 - [Adding Approvers to Service Requests](#)
 - [Removing Approvers from Service Requests](#)
- [Editing Request Tasks](#)

Service Management

Service management is provided via the following collections:

Path	Description
/api/service_catalogs	Service Catalogs
/api/service_templates	Service Templates
/api/services	Services
/api/service_requests	Service Requests

Service Catalogs

Full CRUD is available on service catalogs.

- [Creating Service Catalogs](#)
- [Editing Service Catalogs](#)
- [Assigning/Unassigning Service Templates from Service Catalogs](#)
- [Ordering Services](#)
- [Refresh Service Dialog Fields](#)

- [Deleting Service Catalogs](#)

Creating Service Catalogs

`/api/service_catalogs`

Creating service catalog is provided via the following actions:

Method	Versions	Comment
create	2.0.0	Preferred
add	since v1.0	

Example of creating service catalog with *create* method:

POST `/api/service_catalogs`

```
{
  "action" : "create",
  "resource" : {
    "name" : "Sample Service Catalog",
    "description" : "Description of Sample Service Catalog",
    "service_templates" : [
      { "href" : "http://localhost:3000/api/service_templates/3" },
      { "href" : "http://localhost:3000/api/service_templates/4" }
    ]
  }
}
```

Optionally without the create action signature:

```
{
  "name" : "Sample Service Catalog",
  "description" : "Description of Sample Service Catalog",
  "service_templates" : [
    { "href" : "http://localhost:3000/api/service_templates/3" },
    { "href" : "http://localhost:3000/api/service_templates/4" }
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Service Catalogs.

Editing Service Catalogs

Editing a service catalog via the *edit* action as follows:

POST `/api/service_catalogs/:id`

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Description of Third Service Catalog"
  }
}
```

Assigning Service Templates

Assign and Unassigning service templates to service catalogs is done via the *assign* and *unassign* actions as follows:

```
POST /api/service_catalogs/:id
```

```
{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/6" }
  ]
}
```

```
POST /api/service_catalogs/:id
```

```
{
  "action" : "unassign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/1" },
    { "href" : "http://localhost:3000/api/service_templates/2" }
  ]
}
```

Ordering Services Service Templates

Ordering services is done by triggering an *order* action on service templates in a service catalog as follows:

```
POST /api/service_catalogs/:id/service_templates
```

```
{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001"
  }
}
```

Note: If you need to include a password in the call to the Ordering Services, you can use the `password::dialog_password` field to supply the password as a parameter. This password is passed as a `string` datatype.

```
{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001",
    "password::dialog_password" : "password"
  }
}
```

Refresh Service Dialog Fields

Refreshing dialog fields is done via a *refresh_dialog_fields* action as shown in the following cases:

Refresh Dialog Fields of Service Templates Service Dialogs

Refreshing dialog fields for a service template service dialogs is done as follows:

```
POST /api/service_catalogs/:id/service_templates
```

```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
```

```

    "href" : "http://localhost:3000/api/service_templates/3",
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
    "fields" : [ "my_field1", "my_field3" ]
  }
}

```

Refresh Dialog Fields of Service Dialogs

Refreshing dialog fields for a service dialog is done as follows:

POST /api/service_dialogs/:id

```

{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
    "fields" : [ "my_field1", "my_field3" ]
  }
}

```

Deleting Service Catalogs

Single service catalogs can simply be deleted via the DELETE method as follows:

DELETE /api/service_catalogs/:id

Multiple service catalogs can also be deleted with a single request as follows:

POST /api/service_catalogs

```

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_catalogs/11" },
    { "href" : "http://localhost:3000/api/service_catalogs/12" },
    { "href" : "http://localhost:3000/api/service_catalogs/13" }
  ]
}

```

Service Templates

Management of Service Templates is available via the following collection:

/api/service_templates

Full CRUD is available on service templates.

- [Creating Service Templates](#)
- [Editing Service Templates](#)
- [Deleting Service Templates](#)

Creating Service Templates

Service Templates can be created via a POST to the `/api/service_templates` collection:

`/api/service_templates`

```
{
  "name" : "Atomic Service Template",
  "service_type" : "atomic",
  "prov_type" : "amazon",
  "display" : "false",
  "config_info" : {
    "miq_request_dialog_name" : "Dialog_Name",
    "placement_auto" : [true, 1],
    "number_of_vms" : [1, "1"],
    "src_vm_id" : [11, "test_vm"],
    "vm_name" : "AtomicVMName",
    "schedule_type" : ["immediately", "Immediately on Approval"],
    "instance_type" : [21, "instance_flavor"],
    "src_ems_id" : [5, "test_provider"],
    "provision" : {
      "fqname" : "/Sample/System/ProvisionEndpoint",
      "dialog_id" : "21"
    },
    "retirement" : {
      "fqname" : "/Sample/System/RetirementEndpoint",
      "dialog_id" : "22"
    }
  }
}
```

Editing Service Templates

Editing a single service template as follows:

`POST /api/service_templates/:id`

```
{
  "action" : "edit",
  "resource" : {
    "name" : "svc_template_01",
    "description" : "This is a sample service template"
  }
}
```

Deleting Service Templates

Deleting single service template via the DELETE method:

`DELETE /api/service_templates/:id`

or via the *delete* action as follows:

`POST /api/service_templates/:id`

```
{
  "action" : "delete"
}
```

Multiple service templates can also be deleted in bulk as follows:

`POST /api/service_templates`


```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/11" },
    { "href" : "http://localhost:3000/api/service_templates/12" }
  ]
}
```

Services ===

Accessing services is via the **services** collection as follows:

/api/services

or addressing an individual Service via the **services** resource:

/api/services/:id

Creating Services =====

In general, services are created once ordered from service catalogs.

Service resources can also be created as follows:

POST /api/services

```
{
  "action" : "create",
  "resource" : {
    "name" : "test_service_name",
    "description" : "Test Service",
    "parent_service" : { "href" : "http://localhost:3000/api/services/15" }
  }
}
```

In the above example *parent_service* is an *href* or *id* reference to an existing service.

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Services.

Service Actions =====

Once services are ordered from a catalog, they can be:

- [Edited](#) (Basic)
- [Retired now](#)
- [Retired in future](#)
- [Reconfigured](#)
- [Started](#)
- [Stopped](#)
- [Suspended](#)
- [Deleted](#)

Additional Service actions include:

- [Adding and Removing Resources from Services](#)
- [Adding Provider VMs to Services](#)
- [Requesting a Chargeback Report for a Service](#)

Other than service creates and edits, service actions return the action response as in the following example:

Request:

```
POST /api/services/12
```

```
{
  "action" : "start"
}
```

Response:

```
{
  "success": true,
  "message": "Service id:12 name:'SampleService' starting",
  "task_id": "62",
  "task_href": "http://localhost:3000/api/tasks/62",
  "href": "http://localhost:3000/api/services/12"
}
```

Service Editing

```
POST /api/services/:id
```

```
{
  "action" : "edit",
  "resource" : {
    "name" : "Updated Service Name",
    "description" : "Updated Service Description"
  }
}
```

Adding and Removing Resources from Services

Adding and removing resources from services is made available with the following actions:

Action
add_resource
remove_resource
remove_all_resources

The following example adds a single resource to a service:

```
POST /api/services/:id
```

```
{
  "action" : "add_resource",
  "resource" : { "resource" : { "href" : "http://localhost:3000/api/vms/11" } }
}
```

The following example bulk adds resources to multiple services:

```
POST /api/services
```

```
{
  "action" : "add_resource",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhost:3000/api/vms/11" }
    },
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhost:3000/api/vms/12" }
    },
    {
      "href" : "http://localhost:3000/api/services/4",
      "resource" : { "href" : "http://localhost:3000/api/vms/21" }
    },
    {
      "href" : "http://localhost:3000/api/services/5",
      "resource" : { "href" : "http://localhost:3000/api/vms/22" }
    }
  ]
}
```

The following example removes a single resource from a service:

POST /api/services/:id

```
{
  "action" : "remove_resource",
  "resource" : { "resource" : { "href" : "http://localhost:3000/api/vms/11" } }
}
```

The following example removes resources in bulk from multiple services:

POST /api/services

```
{
  "action" : "remove_resource",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhost:3000/api/vms/11" }
    },
    {
      "href" : "http://localhost:3000/api/services/3",
      "resource" : { "href" : "http://localhost:3000/api/vms/12" }
    },
    {
      "href" : "http://localhost:3000/api/services/4",
      "resource" : { "href" : "http://localhost:3000/api/vms/21" }
    },
    {
      "href" : "http://localhost:3000/api/services/5",
      "resource" : { "href" : "http://localhost:3000/api/vms/22" }
    }
  ]
}
```

The following example removes all resources from a single service:

POST /api/services/:id

```
{
  "action" : "remove_all_resources"
}
```

The following example removes all resources in bulk from multiple services:

POST /api/services

```
{
  "action" : "remove_all_resources",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/11" },
    { "href" : "http://localhost:3000/api/services/12" },
    { "href" : "http://localhost:3000/api/services/13" }
  ]
}
```

Adding Provider VMs to Services

For cases where VMs are created externally, i.e. via Ansible Playbooks, this action allows the system to identify the VMs created and link those VMs back to the service.

POST /api/services/:id

```
{
  "action" : "add_provider_vms",
  "resource" : {
    "provider" : { "href" : "http://localhost:3000/api/providers/11" }
    "uid_ems" : [
      "provider_vm_id1",
      "provider_vm_id2",
      ...
    ]
  }
}
```

Requesting a Chargeback Report for a Service

This action will request a chargeback report for the service.

POST /api/services/:id

```
{
  "action" : "queue_chargeback_report"
}
```

The queue_chargeback_report action will return an action response which includes the task reference as shown here:

```
{
  "success": true,
  "message": "Queued chargeback report generation for Service id:1",
  "name": "test_service1",
  "task_id": "39",
  "task_href": "http://localhost:3000/api/tasks/39"
}
```

The task_href can be queried and upon completion of the task, the chargeback report for the service can be queried as follows:

GET /api/services/:id?attributes=chargeback_report

Service Retiring (now)

POST /api/services/:id

```
{
  "action" : "request_retire"
}
```

Service Retiring (future)

POST /api/services/:id

```
{
  "action" : "request_retire",
  "resource" : { "date" : "10/30/2015", "warn" : "5" }
}
```

Service Reconfiguring

When a service is defined with a Reconfigure Automation Entrypoint, the *reconfigure* action is shown as a valid action for that service.

This action takes any of the dialog fields the service supports during reconfiguration.

POST /api/services/:id

```
{
  "action" : "reconfigure",
  "resource" : {
    "field1" : "updated field1 value",
    "field2" : "updated field2 value",
    ...
  }
}
```

Starting Services

POST /api/services/:id

```
{
  "action" : "start"
}
```

Stopping Services

POST /api/services/:id

```
{
  "action" : "stop"
}
```

Suspending Services

POST /api/services/:id

```
{
  "action" : "suspend"
}
```

Deleting Services

DELETE /api/services/:id

Or via the delete action for deleting multiple services:

```
POST /api/services
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/101",
    { "href" : "http://localhost:3000/api/services/102",
    { "href" : "http://localhost:3000/api/services/103"
    ...
  ]
}
```

Service Requests ===

- [Querying Service Requests](#)
- [Approving or Denying Service Requests](#)
- [Adding Approvers to Service Requests](#)
- [Removing Approvers to Service Requests](#)
- [Editing Request Tasks](#)

Querying Service Requests

Once services are ordered from a catalog, service requests can be monitored as follows:

```
GET /api/service_requests/:id
```

Note:

The request tasks of service requests can be queried by expanding the request_tasks as follows:

```
GET /api/service_requests/:id?expand=request_tasks
```

or by fetching them directly as follows:

```
GET /api/service_requests/:id/request_tasks?expand=resources
```

Approving or Denying Service Requests

If a service request is not auto approved, it can be approved or denied via the actions *approve* and *deny* on the service request resource.

Approving a Service Request

```
POST /api/service_requests/:id
```

```
{
  "action" : "approve",
  "reason" : "reason for approval"
}
```

Denying a Service Request

```
POST /api/service_requests/:id
```

```
{
  "action" : "deny",
```

```
"reason" : "reason for denial"
}
```

Adding Approvers to Service Requests

Approvers can be added to service requests with the *add_approver* action. Approvers can be specified by *user_id* or *user* reference including an *href* or *id* of the approver.

POST /api/service_requests/:id

```
{
  "action" : "add_approver",
  "resource" : {
    "user_id" : 11
  }
}
```

Or one of the following signatures:

```
{
  "action" : "add_approver",
  "resource" : {
    "user" : { "id" : 11 }
  }
}

{
  "action" : "add_approver",
  "resource" : {
    "user" : { "href" : "http://localhost:3000/api/users/11" }
  }
}
```

Approvers can also be added in bulk as follows:

POST /api/service_requests

```
{
  "action" : "add_approver",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_requests/101",
      "user_id" : "11"
    },
    {
      "href" : "http://localhost:3000/api/service_requests/102",
      "user" : { "id" : 11 }
    },
    {
      "href" : "http://localhost:3000/api/service_requests/103",
      "user" : { "href" : "http://localhost:3000/api/users/11" }
    }
  ]
}
```

Removing Approvers from Service Requests

Approvers can be removed from service requests with the *remove_approver* action. Approvers can be specified by *user_id* or *user* reference including an *href* or *id* of the approver.

POST /api/service_requests/:id

```
{
  "action" : "remove_approver",
  "resource" : {
    "user_id" : "11"
  }
}
```

Approvers can also be removed in bulk as follows:

POST /api/service_requests

```
{
  "action" : "remove_approver",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_requests/101",
      "user_id" : "11"
    },
    {
      "href" : "http://localhost:3000/api/service_requests/102",
      "user" : { "id" : "11" }
    },
    {
      "href" : "http://localhost:3000/api/service_requests/103",
      "user" : { "href" : "http://localhost:3000/api/users/11" }
    }
  ]
}
```

Editing Request Tasks

Request tasks of Service Requests can be updated as follows:

POST /api/service_requests/:id/request_tasks/:request_task_id

```
{
  "action" : "edit",
  "resource" : {
    "options" : {
      "request_param_a" : "value_a",
      "request_param_b" : "value_b"
    }
  }
}
```

- [Service Queries](#)
 - [Querying Pictures pertaining to services](#)
 - [Querying Provision Dialogs on services and service requests](#)
 - [Querying Resource Actions](#)
 - [Querying Service and Provision Dialogs](#)
 - [Querying Service Dialogs](#)
 - [Querying Users that ordered services](#)
 - [Querying VMs of Services](#)
 - [Querying Orchestration Stacks of Services](#)

Service Queries

Queries to services and related service dialogs are provided via the following collections:

Path	Description
/api/services	Services
/api/service_requests	Service Requests
/api/service_templates	Service Templates
/api/service_dialogs	Service Dialogs

- [Querying Pictures](#)
- [Querying Provision Dialogs](#)
- [Querying Resource Actions](#)
- [Querying Service and Provision Dialogs](#)
- [Querying Service Dialogs](#)
- [Querying Users](#)
- [Querying VMs](#)
- [Querying Orchestration Stacks](#)

Querying Pictures pertaining to services

Access to pictures for service related collections is made available with the *picture* virtual attribute and related *image_href* attribute.

Querying all resources:

```
GET /api/services?expand=resources&attributes=picture,picture.image_href
```

```
GET /api/service_requests?expand=resources&attributes=picture,picture.image_href
```

```
GET /api/service_templates?expand=resources&attributes=picture,picture.image_href
```

Querying specific resources:

```
GET /api/services/:id?attributes=picture,picture.image_href
```

```
GET /api/service_requests/:id?attributes=picture,picture.image_href
```

```
GET /api/service_templates/:id?attributes=picture,picture.image_href
```

Querying Provision Dialogs on services and service requests

Querying provision dialogs on ordered services and related service_requests is provided by the *provision_dialog* virtual attribute:

```
GET /api/services/:id?attributes=provision_dialog
```

```
GET /api/service_requests/:id?attributes=provision_dialog
```

Querying Resource Actions

Querying resource_actions is provided as a subcollection of service_templates as follows:

```
GET /api/service_templates/:id/resource_actions
```

This allows queries like:

```
GET /api/service_templates/:id/resource_actions
GET /api/service_templates/:id/resource_actions/:resource_action_id
GET /api/service_templates/:id/resource_actions?expand=resources
GET /api/service_templates/:id/resource_actions?
expand=resources&filter[]=action='Provision'
```

Querying Service and Provision Dialogs

Access to service and provision dialogs is provided by the primary collections:

/api/service_dialogs
/api/provision_dialogs

Querying Service Dialogs

Service dialogs can be queried directly from the primary collection:

```
GET /api/service_dialogs
GET /api/service_dialogs/:id
GET /api/service_dialogs/:id?resource_action_id=:ra_id
                                &target_id=:target_id
                                &target_type=:target_type
```

Note:

Above :target_type would be a valid resource type containing the service dialog like *service_template* or *service*.

Or from the Service Template where they are defined and from the related ordered services as follows:

```
GET /api/service_templates/:id/service_dialogs
GET /api/service_templates/:id/service_dialogs/:service_dialog_id

GET /api/services/:id/service_dialogs
GET /api/services/:id/service_dialogs/:service_dialog_id
```

Querying Users that ordered services

This is made available via the *user* virtual attribute which reflect the user object.

```
GET /api/services/:id?attributes=user
GET /api/service_requests/:id?attributes=user
```

Optionally, specific user attribute can also be queried as follows:

```
GET /api/services/:id?attributes=user.name,user.email
GET /api/services/:id?attributes=user.name,user.email
```

Querying VMs of Services

This is made available via the *vm*s subcollection of services as follows:

```
GET /api/services/:id/vms
GET /api/services/:id?expand=vms
```

It is also possible to fetch additional attributes for the vms of a service all within the same request by using the *vm*s. prefix in the attributes specification as follows:

```
GET /api/services/:id?expand=vms&attributes=vms.num_cpu
```

Accessing decorators for VM resources is also made available here as follows:

```
GET /api/services/:id?expand=vms
    &attributes=vms.num_cpu
&decorators=vms.supports_console?,vms.supports_cockpit?,vms.supports_vnc_console?
```

Querying Orchestration Stacks of Services

This is made available via the *orchestration_stacks* subcollection of services as follows:

```
GET /api/services/:id/orchestration_stacks
```

Fetching the orchestration stacks in addition to the service as follows:

```
GET /api/services/:id?expand=orchestration_stacks
```

Accessing a specific orchestration stack of a service as follows:

```
GET /api/services/:id/orchestration_stacks/:orchestration_stack_id
```

- [Service Custom Actions](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)

Service Custom Actions

With button groups, custom buttons and related dialogs defined on Service Templates, the REST API provides visibility to custom buttons on service templates and related ordered services.

This is made available with the *custom_actions* and *custom_action_buttons* virtual attributes on *service_templates* and *services*.

- *custom_actions* returns the custom buttons as layed out in the UI on the service. The information returned includes any Resource Actions and related Dialogs for the custom button.
- *custom_action_buttons* returns the array of custom buttons without the custom button groups hierarchy or details on related dialogs.

Fetching custom actions and custom action buttons for a *service_template* or a deployed service

```
GET /api/service_templates/:id?attributes=custom_actions,custom_action_buttons
```

```
GET /api/services/:id?attributes=custom_actions,custom_action_buttons
```

Example Service Template with custom buttons:

Request:

```
GET /api/service_templates/7?attributes=custom_actions,custom_action_buttons
```

Response:

```
{
  "href": "http://localhost:3000/api/service_templates/7",
```

```

    "id": "7",
    "name": "SampleServiceCatalogItem2",
    "description": "Sample Service Catalog Item 2",
    "guid": "47edc73a-4cef-11e5-8525-000a27020067",
    "options": {
      "button_order": [
        "cb-1",
        "cb-2",
        "cbg-43",
        "cb-3"
      ]
    },
    "created_at": "2015-08-27T19:10:29Z",
    "updated_at": "2015-09-16T18:24:13Z",
    "display": true,
    "evm_owner_id": "1",
    "service_type": "atomic",
    "prov_type": "generic",
    "service_template_catalog_id": "7",
    "long_description": "",
    "tenant_id": "1",
    "custom_actions": {
      "buttons": [
        {
          "id": "1",
          "guid": "3d5975fe-57bb-11e5-8d22-000a27020067",
          "description": "Rescan Service Offering",
          "applies_to_class": "ServiceTemplate",
          "options": {
            "button_image": 2,
            "display": true
          },
          "userid": "admin",
          "created_on": "2015-09-10T12:55:40Z",
          "updated_on": "2015-09-16T14:28:17Z",
          "name": "Rescan",
          "visibility": {
            "roles": [
              "_ALL_"
            ]
          },
          "applies_to_id": "7",
          "resource_action": {
            "id": "7",
            "resource_id": "1",
            "resource_type": "CustomButton",
            "created_at": "2015-09-10T12:55:40Z",
            "updated_at": "2015-09-16T14:28:17Z",
            "ae_namespace": "SYSTEM",
            "ae_class": "PROCESS",
            "ae_instance": "Request",
            "ae_attributes": {
              "key1": "val1",
              "key2": "val2",
              "request": "InspectMe"
            }
          }
        }
      ],
      {
        "id": "2",
        "guid": "7c900d18-57c1-11e5-8d22-000a27020067",
        "description": "Ping Service Status",
        "applies_to_class": "ServiceTemplate",
        "options": {

```

```

        "button_image": 3,
        "display": true
    },
    "userid": "admin",
    "created_on": "2015-09-10T13:40:23Z",
    "updated_on": "2015-09-16T14:16:21Z",
    "name": "Ping",
    "visibility": {
        "roles": [
            "_ALL_"
        ]
    },
    "applies_to_id": "7",
    "resource_action": {
        "id": "8",
        "resource_id": "2",
        "resource_type": "CustomButton",
        "created_at": "2015-09-10T13:40:23Z",
        "updated_at": "2015-09-16T14:16:21Z",
        "ae_namespace": "SYSTEM",
        "ae_class": "PROCESS",
        "ae_instance": "Request",
        "ae_attributes": {
            "Class": "System",
            "rel2": "//ManageIQ/System/Request/InspectMe",
            "request": "InspectMe"
        }
    }
},
],
"button_groups": [
    {
        "id": "43",
        "name": "MyButtonGroup|ServiceTemplate|7",
        "description": "My Custom Buttons",
        "created_on": "2015-09-10T13:41:20Z",
        "updated_on": "2015-09-15T14:53:49Z",
        "guid": "9e4a9900-57c1-11e5-8d22-000a27020067",
        "set_data": {
            "button_order": [
                3
            ],
            "button_image": 1,
            "display": true,
            "applies_to_class": "ServiceTemplate",
            "applies_to_id": "7",
            "group_index": 1
        },
        "owner_type": "ServiceTemplate",
        "owner_id": "7",
        "buttons": [
            {
                "id": "3",
                "guid": "92ac7d88-5bb9-11e5-b2f7-000a27020067",
                "description": "Simulate Button Dialog",
                "applies_to_class": "ServiceTemplate",
                "options": {
                    "button_image": 12,
                    "display": true
                },
                "userid": "admin",
                "created_on": "2015-09-15T14:53:49Z",
                "updated_on": "2015-09-16T14:28:50Z",
                "name": "simulate",

```

```

"visibility": {
  "roles": [
    "_ALL_"
  ]
},
"applies_to_id": "7",
"resource_action": {
  "id": "12",
  "dialog_id": "2",
  "resource_id": "3",
  "resource_type": "CustomButton",
  "created_at": "2015-09-15T14:53:49Z",
  "updated_at": "2015-09-16T14:28:50Z",
  "ae_namespace": "SYSTEM",
  "ae_class": "PROCESS",
  "ae_instance": "Request",
  "ae_attributes": {
    "request": "InspectMe"
  },
},
"dialog": {
  "description": "Test Dialog 2",
  "buttons": "submit, cancel",
  "label": "TestDialog2",
  "dialog_tabs": [
    {
      "description": "VM Sizing",
      "display": "edit",
      "label": "VmSizing",
      "position": 0,
      "dialog_groups": [
        {
          "description": "Memory",
          "display": "edit",
          "label": "Memory",
          "position": 0,
          "dialog_fields": [
            {
              "name": "DynamicMemory",
              "description": "Amount of Memory for VM is Dynamic",
              "type": "DialogFieldCheckBox",
              "display": "edit",
              "display_method_options": {
            },
              "required_method_options": {
            },
              "default_value": "f",
              "values_method_options": {
            },
              "options": {
            },
              "label": "DynamicMemory",
              "position": 0,
              "dynamic": false,
              "read_only": false,
              "resource_action": {
                "resource_type": "DialogField",
                "ae_attributes": {
            }
              }
            }
          ]
        }
      ]
    }
  ]
}

```

```

    ]
  }
}
]
}
],
"custom_action_buttons": [
  {
    "id": "1",
    "guid": "3d5975fe-57bb-11e5-8d22-000a27020067",
    "description": "Rescan Service Offering",
    "applies_to_class": "ServiceTemplate",
    "options": {
      "button_image": 2,
      "display": true
    },
    "userid": "admin",
    "created_on": "2015-09-10T12:55:40Z",
    "updated_on": "2015-09-16T14:28:17Z",
    "name": "Rescan",
    "visibility": {
      "roles": [
        "_ALL_"
      ]
    },
    "applies_to_id": "7"
  },
  {
    "id": "2",
    "guid": "7c900d18-57c1-11e5-8d22-000a27020067",
    "description": "Ping Service Status",
    "applies_to_class": "ServiceTemplate",
    "options": {
      "button_image": 3,
      "display": true
    },
    "userid": "admin",
    "created_on": "2015-09-10T13:40:23Z",
    "updated_on": "2015-09-16T14:16:21Z",
    "name": "Ping",
    "visibility": {
      "roles": [
        "_ALL_"
      ]
    },
    "applies_to_id": "7"
  },
  {
    "id": "3",
    "guid": "b9afcdfa-57c1-11e5-8d22-000a27020067",
    "description": "Simulate Button Dialog",
    "applies_to_class": "ServiceTemplate",
    "options": {
      "button_image": 3,
      "display": true
    },
    "userid": "admin",
    "created_on": "2015-09-10T13:42:06Z",
    "updated_on": "2015-09-16T14:27:30Z",
    "name": "Simulate",
    "visibility": {
      "roles": [

```

```

        "_ALL_"
    ]
},
"applies_to_id": "7"
}
}
"actions": [
{
    "name": "edit",
    "method": "post",
    "href": "http://localhost:3000/api/service_templates/7"
},
{
    "name": "delete",
    "method": "delete",
    "href": "http://localhost:3000/api/service_templates/7"
}
]
}
}

```

When service templates define custom buttons, the ordered services are extended with custom actions which can be triggered via a POST action.

Note that the custom actions are exposed as actions on the deployed services, not the service_templates. The action name to trigger the action is the lower-cased name of the custom button.

As an example, here's a service without custom buttons:

```

{
    "href": "http://localhost:3000/api/services/91",
    "id": "91",
    "name": "new_service_91",
    "description": "This is a description for the new service 91",
    ...
    "actions": [
        {
            "name": "edit",
            "method": "post",
            "href": "http://localhost:3000/api/services/91"
        },
        {
            "name": "request_retire",
            "method": "post",
            "href": "http://localhost:3000/api/services/91"
        },
        {
            "name": "delete",
            "method": "delete",
            "href": "http://localhost:3000/api/services/91"
        }
    ]
}
}

```

Here is an example of a service with custom buttons:

```

{
    "href": "http://localhost:3000/api/services/92",
    "id": "92",
    "name": "enhanced_service_92",
    "description": "This is a description for the enhanced service 92",
    "guid": "24c193a2-57c8-11e5-9332-000a27020067",
    "service_template_id": "7",
    "options": {

```



```

    "button_order": [
      "cb-1",
      "cb-2",
      "cb-3"
    ],
    "dialog": {
      "dialog_DynamicMemory": "t"
    }
  },
  "display": true,
  "created_at": "2015-09-10T14:28:02Z",
  "updated_at": "2015-09-10T14:28:02Z",
  "evm_owner_id": "1",
  "miq_group_id": "1",
  "actions": [
    {
      "name": "edit",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "rescan",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "ping",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    },
    {
      "name": "simulate",
      "method": "post",
      "href": "http://localhost:3000/api/services/92"
    }
  ]
}

```

Here, the *rescan*, *ping* and *simulate* are custom buttons for the service and are returned after the common actions.

For example, to request the *rescan* custom action shown above, this is simply a POST action as follows:

Request:

POST /api/services/92

```

{
  "action" : "rescan"
}

```

Response:

Responses from custom action are the normal action responses from the REST API:

```
{
  "success": true,
  "message": "Invoked custom action rescan for services id: 92",
  "href": "http://localhost:3000/api/services/92"
}
```

- [Service Dialogs](#)
 - [Creating Service Dialogs](#)
 - [Editing Service Dialogs](#)
 - [Copying Service Dialogs](#)
 - [Refreshing Dialog Fields of Service Dialogs](#)
 - [Deleting Service Dialogs](#)

Service Dialogs

Management of Service Dialogs is provided via the collection:

`/api/service_dialogs`

The following actions are provided on service_dialogs resources:

- [Creating Service Dialogs](#)
- [Editing Service Dialogs](#)
- [Copying Service Dialogs](#)
- [Refreshing Dialog Fields of Service Dialogs](#)
- [Deleting Service Dialogs](#)

Creating Service Dialogs

Service Dialogs can be created via a POST to the *service_dialogs* collection or via the **create** action signature which also allows creation of multiple service_dialogs in a single request.

POST `/api/service_dialogs`

```
{
  "description" : "Dialog",
  "label" : "dialog_label",
  "dialog_tabs" : [
    {
      "description" : "Dialog Tab",
      "position" : 0,
      "label" : "dialog_tab_label",
      "dialog_groups" : [
        {
          "description" : "Dialog Group",
          "label" : "group_label",
          "dialog_fields" : [
            {
              "name" : "A Dialog Field",
              "label" : "dialog_field_label"
            }
          ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

or creating multiple Service Dialogs:

```

{
  "action" : "create",
  "resources" : [
    { "description" : "Dialog 1", "label" : "dialog_1_label", ... },
    { "description" : "Dialog 2", "label" : "dialog_2_label", ... },
    ...
  ]
}

```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Service Dialogs.

Editing Service Dialogs

Service Dialogs can be updated via the *edit* action as follows:

POST /api/service_dialogs/:id

```

{
  "action" : "edit",
  "resource" : {
    "label" : "updated_label",
    "content" : {
      "dialog_tabs" : [
        {
          "id" : "11",
          "label" : "updated_tab_label",
          "dialogs_groups" : [
            {
              "id" : "31",
              "dialog_fields" : [
                { "id" : "41" }
              ]
            }
          ]
        }
      ]
    }
  ]
}

```

or editing multiple Service Dialogs in a single request as follows:

POST /api/service_dialogs

```

{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_dialogs/21",
      "label" : "updated_label_1",
      "content" : ...
    },
    {
      "href" : "http://localhost:3000/api/service_dialogs/22",
      "label" : "updated_label_2",
      "content" : ...
    },
  ],
}

```

```

    ...
  ]
}

```

Copying Service Dialogs

Service Dialogs can be copied to create new Service Dialogs via the *copy* action as follows:

```
POST /api/service_dialogs/:id
```

```

{
  "action" : "copy"
}

```

When a label is not specified, the new Service Dialog will be named **Copy of ...** the name of the Service Dialog being copied

Optionally, the label of the new Service Dialog can also be specified as follows:

```

{
  "action" : "copy"
  "resource" : {
    "label" : "test_service_dialog_label"
  }
}

```

Multiple Service Dialogs can also be copied in a single request as follows:

```
POST /api/service_dialogs
```

```

{
  "action" : "copy",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_dialogs/11",
      "label" : "test_service_dialog_label_1"
    },
    {
      "href" : "http://localhost:3000/api/service_dialogs/12",
      "label" : "test_service_dialog_label_2"
    },
    ...
  ]
}

```

Refreshing Dialog Fields of Service Dialogs

Refreshing dialog fields for a Service Dialog is done as follows:

```
POST /api/service_dialogs/:id
```

```

{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "resource_action_id" : "6",
    "target_id" : "12",
    "target_type" : "service_template_ansible_tower",
    "dialog_fields" : {
      "my_field1" : "my field1 value",
      "my_field2" : "my field2 value",
      "my_field3" : "my field3 value",
      ...
    },
  },
}

```

```

    "fields" : [ "my_field1", "my_field3" ]
  }
}

```

Deleting Service Dialogs

Deleting a Service Dialog can be done via the *delete* action as follows:

```
POST /api/service_dialogs/:id
```

```

{
  "action" : "delete"
}

```

or via the DELETE HTTP method as follows:

```
DELETE /api/service_dialogs/:id
```

One can also delete multiple Service Dialogs via the *delete* action on the collection with references to the service dialogs to delete as follows:

```
POST /api/service_dialogs
```

```

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_dialogs/101" },
    { "href" : "http://localhost:3000/api/service_dialogs/102" },
    { "href" : "http://localhost:3000/api/service_dialogs/103" },
    ...
  ]
}

```

- [Service Orders](#)
 - [Querying Service Orders](#)
 - [Creating Service Orders with multiple service requests](#)
 - [Adding Service Requests](#)
 - [Removing Service Requests](#)
 - [Clearing Service Orders](#)
 - [Copying Service Orders](#)
 - [Deleting Service Orders](#)
 - [Ordering Service Orders](#)

Service Orders

The service orders collection and actions are provided for managing Shopping Carts and their related service requests.

```
/api/service_orders
```

Available actions on service_orders instances include:

Action	Description
create	Creating the Shopping Cart
clear	Clearing the Shopping Cart
order	Ordering all service requests in the Shopping Cart
copy	Copying the Shopping Cart

Action	Description
delete	Deleting the Shopping Cart

Some important points on Shopping Carts:

- There is one and only one shopping cart for the authenticated user
- A shopping cart is an `/api/service_orders` resource with a state of **cart**
- Shopping carts can be referenced by their resource id or preferably via the alphanumeric **cart** identifier.
- resource id's need to be used for accessing non-cart service_orders states of **wish** list and **ordered**

Items to add to a shopping cart are *service_requests* that are specified similarly to when services are ordered from a service template.

- [Querying Service Orders](#)
- [Creating Service Orders](#)
- [Adding Service Requests](#)
- [Removing Service Requests](#)
- [Clearing Service Orders](#)
- [Copying Service Orders](#)
- [Deleting Service Orders](#)
- [Ordering Service Orders](#)

Querying Service Orders

All the queries listed here below return the service orders for the authenticated user.

Getting all service_orders:

```
GET /api/service_orders
```

Getting the shopping cart:

```
GET /api/service_orders/cart
```

Getting the shopping cart with expanded service_requests:

```
GET /api/service_orders/cart?expand=service_requests,resources
```

Getting the service requests of the shopping cart:

```
GET /api/service_orders/cart/service_requests
```

Creating Service Orders with multiple service requests

```
POST /api/service_orders
```

```
{
  "service_requests" : [
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "super_vm_a",
      "Basic_Host_Ip" : "192.168.120.101"
    }
  ]
}
```

```

    },
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "super_vm_b",
      "Basic_Host_Ip" : "192.168.120.102"
    },
    ...
  ]
}

```

Creating a wish list service_order:

POST /api/service_orders

```

{
  "state" : "wish",
  "service_requests" : [
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/2",
      "host_name" : "sample_service_order_host_name",
      "host_ip" : "192.168.120.101"
    }
  ]
}

```

Adding Service Requests

Adding service requests to a shopping cart can be done by either of the create methods shown above which will all add the service requests to the existing shopping cart:

POST /api/service_orders

or specifically to the shopping cart or any wish list service_orders via its *id* instead of **cart** as follows:

POST /api/service_orders/cart/service_requests

POST /api/service_orders/:id/service_requests

```

{
  "action" : "add",
  "resources" : [
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "sample_service_order_host_name_2",
      "Basic_Host_Ip" : "192.168.120.102"
    },
    {
      "service_template_href" : "http://localhost:3000/api/service_templates/3",
      "Basic_Host_Name" : "sample_service_order_host_name_3",
      "Basic_Host_Bad_Ip" : "192.168.120.103"
    },
    ...
  ]
}

```

Note: * POST /api/service_orders is atomic, requests are validated for all requests before adding them. * POST /api/service_orders/cart/service_requests will add the requests one by one, the action responses have the **success** field for each requests reflecting the success or failure of each request.

Removing Service Requests

Removing requests from the shopping cart is available via the *remove* action:

```
POST /api/service_orders/cart/service_requests
```

```
{
  "action" : "remove",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_orders/6/service_requests/56" },
    { "href" : "http://localhost:3000/api/service_orders/6/service_requests/57" }
  ]
}
```

Clearing Service Orders

Clearing the shopping cart can be done via the *clear* action:

```
POST /api/service_orders/cart
```

```
{
  "action" : "clear"
}
```

Copying Service Orders

A service order can be copied into a new service order via the *copy* action:

```
POST /api/service_orders/:id
```

```
{
  "action" : "copy",
  "resource" : {
    "name" : "new_service_order_name"
  }
}
```

Service orders can also be copied in bulk as follows:

```
POST /api/service_orders/:id
```

```
{
  "action" : "copy",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_orders/1",
      "name" : "new_service_order_name_1"
    },
    {
      "href" : "http://localhost:3000/api/service_orders/2",
      "name" : "new_service_order_name_2"
    },
    ...
  ]
}
```

Deleting Service Orders

Deleting the shopping cart can be done via the *delete* action:

```
POST /api/service_orders/cart
```

```
{
  "action" : "delete"
}
```

or simply via the DELETE method:

DELETE /api/service_orders/cart

Ordering Service Orders

Ordering the shopping cart is done via the *order* action as follows:

POST /api/service_orders/cart

```
{  
  "action" : "order"  
}
```

- [Setting Ownership](#)
 - [Specifying Ownership](#)
 - [Setting Ownership to single resource](#)
 - [Request:](#)
 - [Response:](#)
 - [Setting Ownership to multiple resources](#)
 - [Request:](#)
 - [Response:](#)

Setting Ownership

Setting ownership of resources is provided with the **set_ownership** action. This action is available for the following resources:

/api/auth_key_pairs
/api/cloud_templates
/api/instances
/api/service_templates
/api/services
/api/templates
/api/vms

- [Specifying Ownership](#)
- [Setting Ownership to single resource](#)
- [Setting Ownership to multiple resources](#)

Specifying Ownership

Specifying ownership in requests is done via the **owner** and **group** specification.

```
{  
  "owner" : { "href" : "http://localhost:3000/api/users/:id" },  
  "group" : { "href" : "http://localhost:3000/api/groups/:id" }  
}
```

Note:

Specifying an owner can be done via **href**, **id**, **name** or **userid**

Note:

Specifying a group can be done via **href**, **id** or **description**

Setting ownership can be done on a specific resource or multiple resources in a single request:

Setting Ownership to single resource

Example setting ownership to a single resource:

Request:

POST /api/vms/320

```
{
  "action" : "set_ownership",
  "resource" : {
    "owner" : { "userid" : "jdoe" },
    "group" : { "description" : "TestGroup" }
  }
}
```

Response:

```
{
  "success": true,
  "message": "setting ownership of vms id 320 to owner: jdoe, group: TestGroup",
  "href": "http://localhost:3000/api/vms/320"
}
```

Setting Ownership to multiple resources

Example setting ownership to multiple resources with a single request:

Request:

POST /api/services

```
{
  "action" : "set_ownership",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/104", "owner" : { "name" :
"John Doe" } },
    { "href" : "http://localhost:3000/api/services/105", "owner" : { "name" :
"John Doe" } },
    { "href" : "http://localhost:3000/api/services/106", "owner" : { "name" :
"John Doe" } }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "setting ownership of services id 104 to owner: John Doe",
      "href": "http://localhost:3000/api/services/104"
    },
    {
      "success": true,
      "message": "setting ownership of services id 105 to owner: John Doe",

```

```

    "href": "http://localhost:3000/api/services/105"
  },
  {
    "success": true,
    "message": "setting ownership of services id 106 to owner: John Doe",
    "href": "http://localhost:3000/api/services/106"
  }
}

```

- [Settings Management](#)
 - [Querying Settings](#)
 - [Updating Settings](#)
 - [Request:](#)
 - [Response:](#)
 - [Deleting Settings](#)
 - [Example:](#)

Settings Management

Management of advanced settings is available on servers, zones and regions. They are provided with the following collection:

/api/settings

for the server processing the API, or for a particular server, zone or region as follows:

/api/servers/:id/settings

/api/zones/:id/settings

/api/regions/:id/settings

The following actions are available with advanced settings:

- [Querying Settings](#)
- [Updating Settings](#)
- [Deleting Settings](#)

Querying Settings

Settings can be queried as follows:

GET /api/servers

GET /api/servers/:id/settings

GET /api/zones/:id/settings

GET /api/regions/:id/settings

Note:

Authorization

- For users with an *EvmGroup-super_administrator* role, the complete advanced settings are returned.
- For users with roles that include the *Configuration Settings* feature, only a whitelisted set of settings are returned.

- Requests from all other users will be rejected with a Forbidden error.

Updating Settings

Updating settings can only be performed by users with an *EvmGroup-super_administrator* role.

Updates to settings is available with the HTTP PATCH method as follows:

```
PATCH /api/servers/:id/settings
```

```
PATCH /api/zones/:id/settings
```

```
PATCH /api/regions/:id/settings
```

The following is an example updating the policy_events's purge_window_size:

Request:

```
PATCH /api/servers/1/settings
```

```
{
  "policy_events" : {
    "purge_window_size" : 500
  }
}
```

Response:

The complete updated settings is returned:

```
{
  "ems": {
    "ems_ansible_tower": {
      "blacklisted_event_names": [
        ],
      ...
    "policy_events" : {
      "history": {
        "keep_policy_events": "6.months",
        "purge_window_size": 1000
      },
      "purge_window_size" : 500
    },
    ...
  }
}
```

Deleting Settings

Deleting settings can only be performed by users with an *EvmGroup-super_administrator* role.

Deleting settings is available with the HTTP DELETE method as follows:

```
DELETE /api/servers/:id/settings
```

```
DELETE /api/zones/:id/settings
```

```
DELETE /api/regions/:id/settings
```

Requests must include an array representing the path of the setting to delete.

Example:

```
DELETE /api/servers/1/settings
```

```
[ "policy_events", "purge_window_size" ]
```

This request deletes the policy_event's purge_window_size setting for /api/servers/1 and its value would revert to its default.

- [Snapshot Management](#)
 - [Snapshot Actions](#)
 - [Querying Snapshots](#)
 - [Creating Snapshots](#)
 - [Reverting Snapshots](#)
 - [Deleting Snapshots](#)

Snapshot Management

Management of snapshots is provided for virtual machines and instances via the snapshots subcollection:

```
/api/vms/:id/snapshots
```

```
/api/instances/:id/snapshots
```

Snapshot Actions

The following snapshot actions are available for the supported resources:

Snapshot Action	Vms	Instances
read	X	X
create	X	X
revert	X	
delete	X	X

- [Querying Snapshots](#)
- [Creating Snapshots](#)
- [Reverting Snapshots](#)
- [Deleting Snapshots](#)

Querying Snapshots

Snapshots of machines are queried via the subcollection of vms and instances as follows:

Querying all snapshots of a VM:

```
GET /api/vms/:id/snapshots
```

Querying all snapshots of an Instance:

```
GET /api/instances/:id/snapshots
```

Filtering, sorting and paging on the subcollection as mentioned on the [Querying](#) page.

For example:

```
GET /api/vms/11/snapshots?expand=resources&filter[]=name=test*
```

One can also fetch a single snapshot by id:

```
GET /api/vms/:id/snapshots/:snapshot_id
```

```
GET /api/instances/:id/snapshots/:snapshot_id
```

Creating Snapshots

Creating snapshots on Vms and Instances is done by posting the request to the snapshots subcollection.

```
POST /api/vms/:id/snapshots
```

```
POST /api/instances/:id/snapshots
```

```
{
  "name" : "name_of_new_snapshot",
  "description" : "Description of new Snapshot",
  "memory" : false
}
```

In the above example, only *name* is required. *memory* can be set to true to have the snapshot include physical memory.

Reverting Snapshots

Reverting snapshots is only available for Vms.

```
POST /api/vms/:id/snapshots/:snapshot_id
```

```
{
  "action" : "revert"
}
```

Deleting Snapshots

Individual snapshots of Vms and Instances can be deleted via the DELETE method as follows:

```
DELETE /api/vms/:id/snapshots/:snapshot_id
```

```
DELETE /api/instances/:id/snapshots/:snapshot_id
```

As well as the *delete* POST action as follows for VMs:

```
POST /api/vms/:id/snapshots/:snapshot_id
```

or instances:

```
POST /api/instances/:id/snapshots/:snapshot_id
```

```
{
  "action" : "delete"
}
```

Snapshots can also be deleted in bulk via the *delete* action on the snapshots subcollection as follows:

```
POST /api/vms/11/snapshots
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/11/snapshots/3" },
    { "href" : "http://localhost:3000/api/vms/11/snapshots/4" },
  ]
}
```

```
{ "href" : "http://localhost:3000/api/vms/11/snapshots/5" }
]
```

As well as for instances:

POST /api/instances/201/snapshots

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/instances/201/snapshots/7" },
    { "href" : "http://localhost:3000/api/instances/201/snapshots/8" }
  ]
}
```

- [Tags & Tagging](#)
 - [Tag Management](#)
 - [Querying Tags](#)
 - [Querying Tags of a Resource](#)
 - [Assigning Tags](#)
 - [Tag specification](#)
 - [Assigning tags to a resource](#)
 - [Response:](#)
 - [Unassigning tags from a resource](#)
 - [Response:](#)
 - [Bulk Assigning Tags](#)
 - [Response:](#)
 - [Bulk Unassigning Tags](#)
 - [Response:](#)

Tags & Tagging

- [Tag Management](#)
- [Querying Tags](#)
- [Querying Tags of a Resource](#)
- [Assigning Tags](#)
- [Tag Specification](#)
- [Assigning Tags to a Resource](#)
- [Unassigning Tags from a Resource](#)
- [Bulk Assigning Tags](#)
- [Bulk Unassigning Tags](#)

Tag Management

Management of Tags is provided in the following forms:

- Querying of tags via the primary collection:

`/api/tags`

- Assigning and Unassigning tags to resources via the tags subcollection of a resource

`/api/:collection/:id/tags`

and posting **assign** or **unassign** actions to the tags subcollection.

Querying Tags

Querying all tags in the system is simply:

GET `/api/tags`

However, that only gives us the href's of the tags:

```
{
  "name": "tags",
  "count": 1385,
  "subcount": 1385,
  "resources": [
    {
      "href": "http://localhost:3000/api/tags/1"
    },
    {
      "href": "http://localhost:3000/api/tags/2"
    },
    {
      "href": "http://localhost:3000/api/tags/3"
    }
    ...
  ]
}
```

Which is not too useful.

Expanding the resources, does provide us with the tag name as follows:

GET `/api/tags?expand=resources`

```
{
  "name": "tags",
  "count": 1385,
  "subcount": 1385,
  "resources": [
    {
      "href": "http://localhost:3000/api/tags/1",
      "id": "1",
      "name": "/managed/roles/change_managers"
    },
    {
      "href": "http://localhost:3000/api/tags/2",
      "id": "2",
      "name": "/managed/roles/operators"
    },
    {
      "href": "http://localhost:3000/api/tags/3",
      "id": "3",
      "name": "/managed/roles/cfme_administrators"
    },
    {
      "href": "http://localhost:3000/api/tags/4",
      "id": "4",

```



```

    "name": "/managed/location"
  },
  {
    "href": "http://localhost:3000/api/tags/5",
    "id": "5",
    "name": "/managed/location/ny"
  }
  ...
]
}

```

For additional details on tags, the **category** and **classification** attributes can be queried on tags as follows:

```
GET /api/tags/5?attributes=category,classification
```

Additionally, a compound attribute called **categorization** is provided to return key information from both **category** and **classification** of tags as follows:

```
GET /api/tags/5?attributes=categorization
```

```

{
  "href": "http://localhost:3000/api/tags/5",
  "id": "5",
  "name": "/managed/location/ny",
  "categorization": {
    "name": "ny",
    "description": "New York",
    "category": {
      "name": "location",
      "description": "Location"
    },
    "display_name": "Location: New York"
  },
}

```

Querying Tags of a Resource

To find out which tags are assigned to a resource, the tags subcollection can be expanded for that particular resource as follows:

```
GET /api/vms/320?expand=tags
```

This will provide the id, href and name of the tags:

```

{
  "href": "http://localhost:3000/api/vms/320",
  "id": "320",
  "vendor": "vmware",
  "name": "aab-vm1",
  "description": "this is a test",
  "raw_power_state": "poweredOn",
  ...
  "tags": [
    {
      "href": "http://localhost:3000/api/vms/320/tags/81",
      "id": "81",
      "name": "/managed/quota_max_cpu/4"
    },
    {
      "href": "http://localhost:3000/api/vms/320/tags/61",
      "id": "61",
      "name": "/managed/cc/001"
    },
  ],
}

```

```

    ...
  ]
}

```

Optionally, one can also query just the tags subcollection of the resource to get additional information about them like the categorization:

GET /api/vms/320/tags?expand=resources&attributes=categorization

```

{
  "name": "tags",
  "count": 1385,
  "subcount": 5,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/320/tags/81",
      "id": "81",
      "name": "/managed/quota_max_cpu/4",
      "categorization": {
        "name": "4",
        "description": "4",
        "category": {
          "name": "quota_max_cpu",
          "description": "Quota - Max CPUs"
        },
        "display_name": "Quota - Max CPUs: 4"
      }
    },
    {
      "href": "http://localhost:3000/api/vms/320/tags/61",
      "id": "61",
      "name": "/managed/cc/001",
      "categorization": {
        "name": "001",
        "description": "Cost Center 001",
        "category": {
          "name": "cc",
          "description": "Cost Center"
        },
        "display_name": "Cost Center: Cost Center 001"
      }
    },
    ...
  ]
}

```

Assigning Tags

Tag management on resources can be done by POSTing **assign** and **unassign** actions to the tags subcollection of resources. Tags can be managed as subcollections of the following primary collections:

Collection
/api/categories
/api/clusters
/api/data_stores
/api/generic_objects
/api/groups
/api/hosts
/api/lans
/api/providers

Collection
/api/resource_pools
/api/services
/api/service_templates
/api/templates
/api/tenants
/api/users
/api/vms

Tag specification

Tags can be specified using one of the following forms:

By Tag category and name:

```
{
  "category" : "department",
  "name"     : "finance"
}
```

By Tag path:

```
{
  "path" : "/managed/department/finance"
}
```

By Tag href:

```
{
  "href" : "http://localhost:3000/api/tags/10"
}
```

Assigning tags to a resource

POST /api/vms/320/tags

```
{
  "action" : "assign",
  "resources" : [
    { "category" : "department", "name" : "finance" },
    { "category" : "cc", "name" : "001" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'cc' name:'001'",

```

```

    "href": "http://localhost:3000/api/vms/320",
    "tag_category": "cc",
    "tag_name": "001",
    "tag_href": "http://localhost:3000/api/tags/91"
  }
}

```

Unassigning tags from a resource

POST /api/vms/320/tags

```

{
  "action": "unassign",
  "resources": [
    { "category": "department", "name": "finance" },
    { "category": "cc", "name": "001" }
  ]
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/vms/320",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
  ]
}

```

Additional Tag Management examples can be found on the main REST API Examples section.

Bulk Assigning Tags

While the general URL for tag assignment via the subcollection as follows:

POST /api/vms/320/tags

allows one to assign multiple tags to a single resource in one call, it does not provide the ability to assign tags to multiple resources in one call.

Bulk assignment of tags is available via the *assign_tags* action to the following collections:

Collection
/api/vms
/api/services

The *assign_tags* action allows adding one or more tags to one or more resources.

POST /api/vms

```
{
  "action" : "assign_tags",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/vms/11",
      "tags" : [
        { "category" : "department", "name" : "finance" }
      ]
    },
    {
      "href" : "http://localhost:3000/api/vms/12",
      "tags" : [
        { "category" : "cc", "name" : "001" }
      ]
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/vms/11",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/vms/12",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
  ]
}
```

Bulk Unassigning Tags

Tags can also be unassigned in bulk from multiple resources for the following collections via the *unassign_tags* action:

Collection
/api/vms
/api/services

POST /api/services

```
{
  "action" : "unassign_tags",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/5",
      "tags" : [
        { "href" : "http://localhost:3000/api/services/5/tags/81" }
      ]
    }
  ],
}
```

```
{
  "href" : "http://localhost:3000/api/services/6",
  "tags" : [
    { "category" : "cc", "name" : "001" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/5",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/81"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'cc' name:'001'",
      "href": "http://localhost:3000/api/services/6",
      "tag_category": "cc",
      "tag_name": "001",
      "tag_href": "http://localhost:3000/api/tags/91"
    }
  ]
}
```

- [Tasks](#)
 - [Querying Tasks](#)
 - [Deleting Tasks](#)

Tasks

Management of Tasks is available via the following endpoint:

/api/tasks

Querying Tasks

Querying all Tasks:

GET /api/tasks

Getting details of a specific Task:

GET /api/tasks/:id

The following actions are available on tasks:

- [Deleting Tasks](#)

Deleting Tasks

Tasks can be deleted via either the **delete** POST action or the DELETE HTTP method.

POST /api/tasks/101

```
{  
  "action" : "delete"  
}
```

or simply:

DELETE /api/tasks/101

Deleting multiple Tasks can be done as follows:

POST /api/tasks

```
{  
  "action" : "delete",  
  "resources" : [  
    { "href" : "http://localhost:3000/api/tasks/101" },  
    { "href" : "http://localhost:3000/api/tasks/102" },  
    ...  
  ]  
}
```

- [Tenant Management](#)
- [Tenant Quota Management](#)
 - [Creating Tenants](#)
 - [Editing Tenants](#)
 - [Deleting Tenants](#)
 - [Creating Tenant Quotas](#)
 - [Editing Tenant Quotas](#)
 - [Deleting Tenant Quotas](#)

Tenant Management

Tenant management including CRUD operations is available via the REST API. Management of tenants is through:

/api/tenants

- [Creating Tenants](#)
- [Editing Tenants](#)
- [Deleting Tenants](#)

Tenant Quota Management

Tenant Quota management is made available via the *quotas* subcollection of tenant resources as follows:

/api/tenants/:id/quotas

- [Creating Tenant Quotas](#)
- [Editing Tenant Quotas](#)
- [Deleting Tenant Quotas](#)

Creating Tenants

Creating a Tenant is done via the **create** action as follows:

```
POST /api/tenants
```

```
{
  "action" : "create",
  "resource" : {
    # New Tenant JSON
  }
}
```

And like other **create** actions, the new tenant data can simply be posted to the /api/tenants URL.

```
POST /api/tenants
```

```
{
  # New Tenant JSON
}
```

For example, creating a new tenant:

```
POST /api/tenants
```

```
{
  "name" : "Test Tenant",
  "description" : "Test Tenant Description",
  "parent" : { "href" : "http://localhost:3000/api/tenants/1" }
}
```

Parent can be specify by *id* or *href*

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Tenants.

Editing Tenants

Editing a Tenant is available via the **edit** action with the **resource** specifying the attributes to update for that tenant.

```
POST /api/tenants/:id
```

```
{
  "action" : "edit" ,
  "resource" : {
    "description" : "Updated Description for Tenant"
  }
}
```

Note:

type is a restricted attribute and cannot be edited.

Deleting Tenants

Deleting a Tenant is done via the delete method:

```
DELETE /api/tenants/:id
```

or via the **delete** action as follows:

```
POST /api/tenants/:id
```

```
{
  "action" : "delete"
```



```
}
```

or by href:

```
POST /api/tenants
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/tenants/:id" }
  ]
}
```

Creating Tenant Quotas

Creating a quota for a specific tenant is done as follows:

```
POST /api/tenants/:id/quotas
```

```
{
  "name" : "cpu_allocated",
  "value" : 1
}
```

Editing Tenant Quotas

```
POST /api/tenants/:id/quotas/:quota_id
```

```
{
  "action" : "edit",
  "resource" : {
    "value" : 5
  }
}
```

Multiple quotas for a tenant can also be edited in a single request as follows:

```
POST /api/tenants/1/quotas
```

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/tenants/1/quotas/5",
      "value" : 11
    },
    { "href" : "http://localhost:3000/api/tenants/1/quotas/6",
      "value" : 12
    },
    ...
  ]
}
```

Deleting Tenant Quotas

Quotas of a tenant can be deleted via the *delete* action as follows:

```
POST /api/tenants/:id/quotas/:quota_id
```

```
{
  "action" : "delete"
}
```

or via the DELETE HTTP method

DELETE /api/tenants/:id/quotas/:quota_id

Multiple quotas of a tenant can be deleted in a single request as shown in the following example:

POST /api/tenants/1/quotas

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/tenants/1/quotas/7" },
    { "href" : "http://localhost:3000/api/tenants/1/quotas/8" },
    ...
  ]
}
```

- [User Management](#)
 - [Querying Users](#)
 - [Creating Users](#)
 - [Editing Users](#)
 - [Editing Authenticated User](#)
 - [Deleting Users](#)

User Management

Management of Users is provided via the following collection:

/api/users

Full CRUD actions on users is available:

- [Querying Users](#)
- [Creating Users](#)
- [Editing Users](#)
- [Editing Authenticated User](#)
- [Deleting Users](#)

Querying Users

Querying all users in the system is simply:

GET /api/users

Getting details on a specific user:

GET /api/users/:id

Creating Users

Users can be created via a POST to the users collection or via the **create** action signature which also allows creation of multiple users in a single request.

POST /api/users

```
{
  "userid" : "testuser",
  "password" : "secret",
  "name" : "Test User",
  "group" : { "id" : "11" }
}
```

Note:

group can be specified via **description**, **href**, or **id**

or creating multiple users:

```
{
  "action" : "create",
  "resources" : [
    { "userid" : "testuser1", "name" : "Test User1", ... },
    { "userid" : "testuser2", "name" : "Test User2", ... },
    ...
  ]
}
```

Note:

Please refer to the [Resource Attributes](#) page for a list of available attributes when creating Users.

Editing Users

POST /api/users/:id

```
{
  "action" : "edit",
  "resource" : {
    "group" : { "id" : "12" }
  }
}
```

or editing multiple users:

POST /api/users

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/users/101",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    {
      "href" : "http://localhost:3000/api/users/102",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    {
      "href" : "http://localhost:3000/api/users/103",
      "group" : { "href" : "http://localhost:3000/api/groups/15" }
    },
    ...
  ]
}
```

Editing Authenticated User

An API Authenticated User can edit some of their own attributes without having general user edit permissions. Editable attributes for the authenticated user are:

password
email
settings

Users can also update their currently selected group with the **set_current_group** action as follows:

POST /api/users/:id

```
{
  "action" : "set_current_group",
  "resource" : {
    "group" : { "description" : "EvmGroup-auditor" }
  }
}
```

Note:

The group reference can be via **description**, **href** or **id**. Also, only a group that the user belongs to can be specified.

Deleting Users

users can be deleted via either the **delete** POST action or via the DELETE HTTP method.

POST /api/users/101

```
{
  "action" : "delete"
}
```

or simply:

DELETE /api/users/101

Deleting multiple users can be done as follows:

POST /api/users

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/users/101" },
    { "href" : "http://localhost:3000/api/users/102" },
    ...
  ]
}
```

- [Virtual Machine Management](#)
 - [Targeting Virtual Machines](#)
 - [Querying Virtual Machines](#)
 - [Starting a Virtual Machine](#)
 - [Stopping a Virtual Machine](#)
 - [Suspending a Virtual Machine](#)
 - [Refreshing a Virtual Machine](#)
 - [Retiring a Virtual Machine](#)
 - [Resetting a Virtual Machine](#)

- [Rebooting guest in Virtual Machine](#)
- [Shutting Down Guest in a Virtual Machine](#)
- [Scanning a Virtual Machine](#)
- [Setting Server of a Virtual Machine](#)
- [Setting Owner of a Virtual Machine](#)
- [Adding an Event to a Virtual Machine](#)
- [Editing a Virtual Machine](#)
- [Deleting a Virtual Machine](#)

Virtual Machine Management

Management of virtual machines adds support of the following actions:

Action	Description
start	Starts a virtual machine
stop	Stops a virtual machine
suspend	Suspends a virtual machine
refresh	Refreshes a virtual machine
request_retire	Retires a virtual machine
reset	Resets a virtual machine
reboot_guest	Reboots guest in virtual machine
shutdown_guest	Shuts down guest in virtual machine
scan	Scans a virtual machine (Perform SmartState Analysis)
set_miq_server	Sets the server of a virtual machine
set_owner	Sets the owner of a virtual machine
add_event	Adding an event to a virtual machine
edit	Edits a virtual machine
delete	Deletes a virtual machine in the appliance

- [Targeting VMs](#)
- [Querying VMs](#)
- [Starting a VM](#)
- [Stopping a VM](#)
- [Suspending a VM](#)
- [Refreshing a VM](#)
- [Retiring a VM](#)
- [Resets a VM](#)
- [Reboots guest in VM](#)
- [Shuts down guest in VM](#)
- [Scanning a VM](#)
- [Setting Server of a VM](#)

- [Setting Owner of a VM](#)
- [Adding an Event to a VM](#)
- [Editing a VM](#)
- [Deleting a VM](#)

Targeting Virtual Machines

These actions can be triggered on individual virtual machine resources:

`/api/vms/:id`

As simply as POSTing the following action to a virtual machine.

```
{
  "action" : "start"
}
```

Requests can also be made on multiple vms by targetting the primary collection:

`/api/vms`

```
{
  "action" : "start",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/11" },
    { "href" : "http://localhost:3000/api/vms/12" },
    ...
  ]
}
```

Querying Virtual Machines

Virtual machines are queried via the primary collection URL:

`/api/vms`

Filtering, sorting and paging as mentioned on the [Querying](#) page.

When querying vms, expanding the resources themselves as well as the following subcollections:

accounts
software

For example:

`GET /api/vms?expand=resources,accounts,software`

Or querying an individual virtual machine

`GET /api/vms/:id?expand=accounts,software`

Starting a Virtual Machine

```
{
  "action" : "start"
}
```

Stopping a Virtual Machine

```
{  
  "action" : "stop"  
}
```

Suspending a Virtual Machine

```
{  
  "action" : "suspend"  
}
```

Refreshing a Virtual Machine

```
{  
  "action" : "refresh"  
}
```

Retiring a Virtual Machine

```
{  
  "action" : "request_retire"  
}
```

Resetting a Virtual Machine

```
{  
  "action" : "reset"  
}
```

Rebooting guest in Virtual Machine

```
{  
  "action" : "reboot_guest"  
}
```

Shutting Down Guest in a Virtual Machine

```
{  
  "action" : "shutdown_guest"  
}
```

Scanning a Virtual Machine

```
{  
  "action" : "scan"  
}
```

Setting Server of a Virtual Machine

```
{  
  "action" : "set_miq_server",  
  "resource" : {  
    "miq_server" : { "href" : "http://localhost:3000/api/servers/5" }  
  }  
}  
  
{  
  "action" : "set_miq_server",  
  "resource" : {  
    "miq_server" : { "id" : "6" }  
  }  
}
```

To remove the server from the virtual machine, pass in an empty reference as follows:

```
{
  "action" : "set_miq_server",
  "resource" : {
    "miq_server" : {}
  }
}
```

Setting Owner of a Virtual Machine

```
{
  "action" : "set_owner",
  "resource" : {
    "owner" : "admin"
  }
}
```

Adding an Event to a Virtual Machine

```
{
  "action" : "add_event",
  "resource" : {
    "event_type" : "...",
    "event_message" : "...",
    "event_time" : "UTC Time"
  }
}
```

Note:

event_time above is optional. If skipped, current time will be used.

Editing a Virtual Machine

Basic information of virtual machines can be edited. This includes the following:

VM Info	Attribute
description	description
custom attributes	custom_1 ... custom_9
parent resource	parent_resource - resource href reference
child resources	child_resources - array of resource href references

Virtual machine resources can be edited as follows:

POST /api/vms/:id

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated VM Description",
    "custom_1" : "custom_attribute_1",
    "parent_resource" : { "href" : "http://localhost:3000/api/vms/11" },
    "child_resources" : [
      { "href" : "http://localhost:3000/api/vms/101" },
      { "href" : "http://localhost:3000/api/vms/102" },
      { "href" : "http://localhost:3000/api/vms/103" },
      { "href" : "http://localhost:3000/api/vms/104" }
    ]
  }
}
```


Virtual machines can also be edited in bulk as follows:

POST /api/vms

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/vms/11",
      "custom_9" : "vm_class_a"
    },
    {
      "href" : "http://localhost:3000/api/vms/12",
      "custom_9" : "vm_class_a"
    },
    {
      "href" : "http://localhost:3000/api/vms/13",
      "custom_9" : "vm_class_a"
    }
  ]
}
```

Deleting a Virtual Machine

```
{
  "action" : "delete"
}
```

Or simply doing the following:

DELETE /api/vms/:id

See the main REST API examples section for additional virtual machine management examples.

Infrastructure Management API Appendices

In this guide you will find appendices for the Infrastructure Automation – Infrastructure Management REST API.

- [Quick Reference](#)
- [Collection Metadata](#)
- [Resource Attributes](#)
- [Provider Types](#)
- [Provision Request Supported Attributes](#)
- [Advanced Provisioning Attributes](#)
- [SOAP to REST Mapping guide](#)
- [Deprecated Types](#)
- [Deprecated Resource Attributes](#)
- [Authentication](#)
- [HTTP Headers](#)
- [Listing and Querying Collections and Sub-Collections](#)
- [Collection Queries:](#)
- [Sub-Collection Queries](#)

This document presents the currently available features and capabilities of the REST API.

Authentication

Type	Mechanism
Basic Authentication	Basic HTTP Authorization with user and password
Token Based Authentication	
- Acquiring Token	/api/auth with Basic Authentication
- Authenticating with Token	X-Auth-Token Header

HTTP Headers

Header	Value
Authorization	Basic base64_encoded(user:password)
X-Auth-Token	Token provided by /api/auth
X-MIQ-Group	Group to authorize externally authenticated user
Accept	application/json
Content-Type	application/json

Listing and Querying Collections and Sub-Collections

Feature	Path	
Listing Available Collections	/api	
Listing Collections	/api/<collection>	
Listing Sub-Collections	/api/<collection>/<id>/<sub-collection>	
Querying Capability	Query Parameters	
Paging	offset, limit	
Sorting	sort_by=attr, sort_order=asc	desc
	Can sort by database columns and SQL friendly virtual attributes	
Filtering	filter[]=...	
Querying by Tag	i.e. by_tag=/department/finance	
Querying by Multiple Tags	i.e. by_tag=/department/finance,/department/engineering	
Expanding Results	expand=<what>, i.e. expand=resources,tags,service_templates,...	
Selecting Attributes	attributes=<attr1>,<attr2>,... i.e. attributes=id,name,type,...	
	Attributes can be:	
	Database columns	
	Virtual attributes	
	Relationships	

[Example Queries](#)

Collection Queries:

For a complete list of supported primary collections please refer to the [Primary Collections](#)

Sub-Collection Queries

When applicable, the following subcollections are available:

Sub-Collection	URL
Tagging	/api/<collection>/:id/tags
Policies	/api/<collection>/:id/policies
Policy Profiles	/api/<collection>/:id/policy_profiles
Service Requests	/api/service_templates/:id/service_requests
Request Tasks	
	/api/service_requests/:id/request_tasks
	/api/automation_requests/:id/request_tasks
	/api/provision_requests/:id/request_tasks
Request Tasks can also be accessed via the tasks alias	
	/api/service_requests/:id/tasks
	/api/automation_requests/:id/tasks
	/api/provision_requests/:id/tasks

- [Collection Metadata](#)
 - [Optional Metadata](#)

Collection Metadata

Collection specific metadata can be queried by sending an HTTP OPTIONS request on the specific collection:

OPTIONS /api/:collection

What gets returned is the following structure:

```
{
  "attributes": [
  ],
  "virtual_attributes": [
  ],
  "relationships": [
  ],
  "subcollections": [
  ],
  "data": {
  }
}
```

Entry	Description
attributes	Physical attributes of the collection
virtual_attributes	Virtual attributes of the collection
relationships	Physical or Virtual associations to other resources
subcollections	Subcollections that can be queried
data	Optional meta-data for the collection

The following is an example for the **groups** collection:

OPTIONS /api/groups

```

{
  "attributes": [
    "created_on",
    "description",
    "group_type",
    "id",
    "sequence",
    "settings",
    "tenant_id",
    "updated_on"
  ],
  "virtual_attributes": [
    "allocated_memory",
    "allocated_storage",
    "allocated_vcpu",
    "custom_1",
    "custom_2",
    "custom_3",
    "custom_4",
    "custom_5",
    "custom_6",
    "custom_7",
    "custom_8",
    "custom_9",
    "href_slug",
    "miq_user_role_name",
    "provisioned_storage",
    "read_only",
    "region_description",
    "region_number",
    "user_count"
  ],
  "relationships": [
    "active_vms",
    "custom_action_buttons",
    "custom_actions",
    "custom_attributes",
    "custom_button_sets",
    "custom_buttons",
    "entitlement",
    "miq_custom_attributes",
    "miq_report_results",
    "miq_reports",
    "miq_templates",
    "miq_user_role",
    "miq_widget_contents",
    "miq_widget_sets",
    "taggings",
    "tags",
    "tenant",
    "users",
    "vms"
  ],
  "subcollections": [
    "tags"
  ],
  "data": {
  }
}

```

Optional Metadata

Most collections do not declare additional metadata as shown by the empty *data* hash entry above. Optional metadata is made available as needed for API clients. They are generally used to provide additional information needed for user-interfaces, help complete or validate data for additional API calls, etc.

Currently, optional metadata is made available for the following collections:

Collection
/api/authentications
/api/clusters
/api/container_deployments
/api/custom_buttons
/api/generic_object_definitions
/api/hosts
/api/providers

There is no enforced structure for the *data* element as each collection may need to provide different types of metadata. The following is a partial meta-data example:

OPTIONS /api/generic_object_definitions

```
{
  "attributes": [
    "created_at",
    "description",
    "id",
    "name",
    "properties",
    "updated_at"
  ],
  "virtual_attributes": [
    "generic_objects_count",
    "href_slug",
    "region_description",
    "region_number"
  ],
  "relationships": [
    "custom_action_buttons",
    "custom_actions",
    "custom_button_sets",
    "custom_buttons",
    "generic_objects",
    "picture"
  ],
  "subcollections": [
    "generic_objects"
  ],
  "data": {
    "allowed_association_types": [
      [
        "Availability Zone",
        "AvailabilityZone"
      ],
      ...
      [
        "VM and Instance",
        "Vm"
      ],
      [
        "Zone",
        "Zone"
      ]
    ]
  }
}
```

```

    ]
  ],
  "allowed_types": [
    [
      "Boolean",
      "boolean"
    ],
    [
      "Date/Time",
      "datetime"
    ],
    [
      "Float",
      "float"
    ],
    [
      "Integer",
      "integer"
    ],
    [
      "String",
      "string"
    ],
    [
      "Time",
      "time"
    ]
  ]
}

```

- [Resource Attributes](#)
 - [Actions](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Alert Definitions](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Authentications](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Category Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Chargeback Rate Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Conditions](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Configuration Script Sources](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Generic Objects](#)
 - [Required attributes:](#)
 - [Required attributes:](#)

- [Optional attributes:](#)
- [Generic Object Definitions](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [Group Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
 - [Filters hash:](#)
- [Orchestration Templates](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [Policies](#)
 - [Required attributes:](#)
- [Provider Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [Role Attributes](#)
 - [Required Attributes:](#)
 - [Optional Attributes:](#)
- [Service Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [Service Catalog Attributes](#)
 - [Optional attributes:](#)
- [Service Dialogs](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [Tag Attributes](#)
 - [Required attributes:](#)
- [Tenant Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)
- [User Attributes](#)
 - [Required attributes:](#)
 - [Optional attributes:](#)

Resource Attributes

This page lists the available attributes when creating resources with the REST API

- [Actions](#)
- [Alert Definitions](#)
- [Authentications](#)
- [Categories](#)
- [Chargeback Rates](#)
- [Conditions](#)

- [Configuration Script Sources](#)
- [Generic Objects](#)
- [Generic Object Definitions](#)
- [Groups](#)
- [Orchestration Templates](#)
- [Policies](#)
- [Providers](#)
- [Roles](#)
- [Services](#)
- [Service Catalogs](#)
- [Service Dialogs](#)
- [Tags](#)
- [Tenants](#)
- [Users](#)

Actions

POST /api/actions

Required attributes:

Attribute	Type	Description
action_type	string	Action Type
description	string	Description (unique)

Optional attributes:

Attribute	Type	Description
name	string	Name
options	hash	Options

Alert Definitions

POST /api/alert_definitions

Required attributes:

Attribute	Type	Description
description	string	Description
db	string	Database
expression	hash	Expression
options	hash	Options

Optional attributes:

Attribute	Type	Description
enabled	boolean	Enable the Alert Definition or not - Defaults to true

Authentications

POST /api/authentications

Required attributes:

Attribute	Type	Description
manager_resource	hash	href of the provider

Optional attributes:

Attribute	Type	Description
type	string	Authentication class

Note:

For a complete list of authentication classes supported and the related attributes for each type, send the *OPTIONS /api/authentications* request.

Category Attributes

POST /api/categories

Required attributes:

Attribute	Type	Description
name	string	Name
description	string	Description

Optional attributes:

Attribute	Type	Description
example_text	string	Detailed description
show	boolean	Show Category flag
single_value	boolean	Single Value flag
syntax	string	Syntax of value, string , integer or boolean

Chargeback Rate Attributes

POST /api/rates

Required attributes:

Attribute	Type	Description
chargeback_rate_id	integer	Reference to parent Chargeback. <code><chargeback_rate></code> can be used instead.
group	string	Group rate belongs to, i.e. cpu , memory , net_io , disk_io , etc.
source	string	The input value for calculation, i.e. allocated , used , etc.

Attribute	Type	Description
chargeback_tiers	Array of hashes	Each tier is represented by hash <code>{“start”: 0.0, “finish”: ‘infinity’, “fixed_rate”: 2.0, “variable_rate”: 0.0}</code>

Optional attributes:

Attribute	Type	Description
description	string	Description of the chargeback rate
enabled	boolean	Rate enabled flag
friendly_rate	string	Friendly name of the rate
metric	string	Metrics, i.e. derived_memory_available , etc.
per_time	string	Measured per time, hourly , daily , weekly , monthly or yearly
per_unit	string	Measured per unit, i.e. megabytes , gigabytes , etc.
chargeback_rate	hash	id or href from api/chargebacks collection.
detail_currency	hash	id or href from api/currencies collection. Details of related currency.
chargeable_field	hash	id or href from api/chargeable_fields collection. Record representing metric. Contains equivalent columns: metric , group , source

Conditions

POST `/api/conditions`

Required attributes:

Attribute	Type	Description
description	string	Condition Description
expression	hash	Expression
modifier	string	Modifier
towhat	string	Class name

Optional attributes:

Attribute	Type	Description
name	string	Condition name

Configuration Script Sources

POST `/api/configuration_script_sources`

Required attributes:

Attribute	Type	Description
manager_resource	hash	href or id of the provider

Optional attributes:

Attribute	Type	Description
description	string	Description

Attribute	Type	Description
name	string	Project Name
related	hash	Related attributes and values

Generic Objects

POST /api/generic_objects

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object
generic_object_definition	hash	Generic object definition to create new generic object for, requires an href or id attribute in the hash

POST /api/generic_object_definitions/:id/generic_objects

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object

Optional attributes:

Attribute	Type	Description
uid	string	Uid string
property_attributes	hash	Attribute/Value pairs to define for the generic object
associations	hash	Hash containing association keys each being an array of resource references

Generic Object Definitions

Required attributes:

Attribute	Type	Description
name	string	Name of the generic object definition

Optional attributes:

Attribute	Type	Description
description	string	Description for the generic object definition
properties	hash	Hash containing the attributes , associations and methods of the generic object definition
picture	hash	Picture to use, either an href or id of the picture resource, or the actual picture including the extension and Base64 encoded content

Group Attributes

POST /api/groups

Required attributes:

Attribute	Type	Description
description	string	Description
role	hash	Role the group is assigned to, requires an href , id or name attribute in the hash
tenant	hash	Tenant the group belongs to, requires an href or id attribute in the hash

Optional attributes:

Attribute	Type	Description
filters	hash	Hash that contains the belongsto and managed arrays of filters to assign to the group
filter_expression	hash	Hash representing the expression used for filtering by Tag

Filters hash:

The **filters** hash can include the following keys:

- **belongsto**
 - Array of tags representing the following types of objects:
 - VMs & Templates
 - Clusters, Datastores, Hosts, Managers & Providers
- **managed**
 - Array of tag arrays representing:
 - My Company Tags

Example:

```
"filters" : {
  "managed" : [
    ["/managed/location/ny"],
    ["/managed/environment/prod"]
  ],
  "belongsto" : [
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|vm/EmsFolder|Alberto-Dev",
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|host/EmsCluster|dev-vc60-cluster"
  ]
}
```

Alternatively, company tags can be represented with an expression via the *filter_expression* instead of the *managed* tag array in the *filters* element as follows:

Example with Tags based on expression:

```
"filters" : {
  "managed" : null,
  "belongsto" : [
    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|vm/EmsFolder|Alberto-Dev",

```

```

    "/belongsto/ExtManagementSystem|dev-vc60/EmsFolder|Datacenters/EmsFolder|dev-vc60-DC/EmsFolder|host/EmsCluster|dev-vc60-cluster"
  ]
},
"filter_expression" : {
  "exp": {
    "and": [
      {
        "CONTAINS": {
          "tag": "managed-location",
          "value": "ny"
        }
      },
      {
        "CONTAINS": {
          "tag": "managed-environment",
          "value": "prod"
        }
      }
    ]
  }
}
}

```

Orchestration Templates

POST /api/orchestration_templates

Required attributes:

Attribute	Type	Description
name	string	Name
content	text	Template content

Optional attributes:

Attribute	Type	Description
type	string	Type
description	string	Description
draft	boolean	true or false
ems_id	integer	Id of the provider
orderable	boolean	true or false

Policies

POST /api/policies

Required attributes:

Attribute	Type	Description
name	string	Name
description	string	Description
mode	string	Mode (i.e. <i>compliance</i>)
towhat	string	Target Class
condition_ids	array	Ids of Conditions to assign to the Policy

Attribute	Type	Description
policy_contents	array	Array of hashes including <i>event_id</i> and <i>actions</i> , an array of hashes comprising of an <i>action_id</i> and <i>opts</i> .

Provider Attributes

POST /api/providers

Required attributes:

Attribute	Type	Description
name	string	Name of the provider
type	string	Provider class
hostname	string	Hostname of the Provider
credentials or connection_configurations	hash or array	Credentials information for the provider either via the credentials hash or connection_configurations array

Optional attributes:

Attribute	Type	Description
api_version	string	API Version for communicating with the Provider
uid_ems	string	Domain for OpenStack provider. Required when “api_version” is “v3”
certificate_authority	string	CA for the Provider
connection_configurations	array	Endpoints for the Provider
credentials	hash	Credentials to use for communicating with the Provider, see Provider Support for examples
host_default_vnc_port_start	integer	Starting VNC port
host_default_vnc_port_end	integer	Ending VNC port
ipaddress	string	IP Address of the Provider
port	string	Port of the Provider
realm	string	Realm of the Provider
security_protocol	string	Security Protocol
url	string	URL of the Provider
zone	hash	In which zone to add provider to, hash needs to include an href or id attribute.

Credentials hash specification attributes are:

Type	Auth Types	Auth Attributes
ManageIQ::Providers::Amazon::CloudManager	default	userid, password
ManageIQ::Providers::Atomic::ContainerManager	default, password, bearer	userid, password
ManageIQ::Providers::Azure::CloudManager	default	userid, password
ManageIQ::Providers::Kubernetes::ContainerManager	default, password, bearer	userid, password
ManageIQ::Providers::Microsoft::InfraManager	default	userid, password

Type	Auth Types	Auth Attributes
ManageIQ::Providers::Openshift::ContainerManager	default, password, bearer	userid, password, auth_key
ManageIQ::Providers::Openstack::InfraManager	default, amqp, ssh_keypair	userid, password, auth_key
ManageIQ::Providers::Openstack::CloudManager	default, amqp	userid, password
ManageIQ::Providers::Redhat::InfraManager	default, metrics	userid, password
ManageIQ::Providers::Vmware::InfraManager	default	userid, password
ManageIQ::Providers::Foreman::Provider	default	userid, password
ManageIQ::Providers::Openstack::Provider	default	userid, password

Role Attributes

POST /api/roles

Required Attributes:

Attribute	Type	Description
name	string	Name of user role

Optional Attributes:

Attribute	Type	Description
features	array	Features to assign to the role. Array of { "identifier" OR "href" OR "id" : ... }
settings	hash	Settings for the role, normally: { "restrictions" : { "vms" : "user" OR "user_or_group" } }

Service Attributes

POST /api/services

Required attributes:

Attribute	Type	Description
name	string	Name

Optional attributes:

Attribute	Type	Description
description	string	Description
orchestration_template	hash	href or id of the Orchestration Template
orchestration_manager	hash	href or id of the Orchestration Provider
parent_service	hash	Parent service href or id reference
type	string	Type of Service

Service Catalog Attributes

POST /api/service_catalogs

Attribute	Type	Description

Attribute	Type	Description
name	string	Name of service catalog
description	string	Description of service catalog

Optional attributes:

Attribute	Type	Description
service_templates	array	Array of Service Template hashes to assign to the new Service Catalog. Hash entries to include the href of the service_templates to assign and must not currently be assigned to any Service Catalog.

Service Dialogs

POST /api/service_dialogs

Required attributes:

Attribute	Type	Description
label	string	Label
dialog_tabs	array	Dialog content - array of dialog tabs

Optional attributes:

Attribute	Type	Description
description	string	Description

Tag Attributes

POST /api/tags

Required attributes:

Attribute	Type	Description
name	string	Name of tag
description	string	Description of tag
category	hash	Category identifier for which the tag is getting added to, requires an href , id or name attribute in the hash

Note:

When adding with a **create** action on the tags subcollection of the category via /api/categories/:id/tags, the category should not be specified in the resource.

Tenant Attributes

POST /api/tenants

Required attributes:

Attribute	Type	Description
name	string	Name of Tenant
parent	hash	Parent tenant identifier for which the sub-tenant is getting added to, requires an href or id attribute in the hash

Optional attributes:

Attribute	Type	Description
description	string	Description of Tenant
domain	string	Domain
logo_file_name	string	Logo file name, must be available via /uploads/ logo_file_name
login_logo_file_name	string	Login Logo file name, must be available via /uploads/ login_logo_file_name
login_text	string	Login text
subdomain	string	Subdomain

User Attributes

POST /api/users

Required attributes:

Attribute	Type	Description
userid	string	Username
password	string	Password
name	string	Full Name
group	hash	Group the user belongs to, requires a description , href or id attribute in the hash

Optional attributes:

Attribute	Type	Description
email	string	E-mail Address

- [Provider Types](#)

Provider Types

The provider types available with earlier appliances have changed with this release. This is what gets specified with the *type* attribute when creating a new provider:

POST /api/providers

```
{
  "name" : "vCenter 5",
  "type" : "ManageIQ::Providers::Vmware::InfraManager",
  "hostname" : "my_vcenter_50",
  ...
}
```

The following table lists the type from previous appliance releases (v2.0.0 of the REST API), and their mapping to the new v2.1.0 types:

v2.0.0 Provider Type	v2.1.0 Provider Type
EmsAmazon	ManageIQ::Providers::Amazon::CloudManager
	ManageIQ::Providers::Atomic::ContainerManager (new)
	ManageIQ::Providers::Azure::CloudManager (new)

v2.0.0 Provider Type	v2.1.0 Provider Type
EmsKubernetes	ManageIQ::Providers::Kubernetes::ContainerManager
EmsMicrosoft	ManageIQ::Providers::Microsoft::InfraManager
	ManageIQ::Providers::Openshift::ContainerManager (new)
EmsOpenstackInfra	ManageIQ::Providers::Openstack::InfraManager
EmsOpenstack	ManageIQ::Providers::Openstack::CloudManager
EmsRedhat	ManageIQ::Providers::Redhat::InfraManager
EmsVmware	ManageIQ::Providers::Vmware::InfraManager

And when *provider-class* is set to **provider**, the following mapping table applies:

v2.0.0 Provider Type	v2.1.0 Provider Type
ProviderForeman	ManageIQ::Providers::Foreman::Provider
ProviderOpenstack	ManageIQ::Providers::Openstack::Provider

- [Provision Request Supported Attributes](#)
 - [Requester Attributes](#)
 - [Custom Attributes](#)
 - [Service Catalog Attributes](#)
 - [Schedule Attributes](#)
 - [Network Attributes](#)
 - [Hardware Attributes](#)
 - [Environment Attributes](#)

Provision Request Supported Attributes

Attribute Class	Provision Request Attribute Group
Requester Attributes	requester
Custom Attributes	vm_fields
Service Catalog Attributes	vm_fields
Schedule Attributes	vm_fields
Network Attributes	vm_fields
Hardware Attributes	vm_fields
Environment Attributes	additional_values

Requester Attributes

Attribute	Type	Description
owner_phone	string	Phone
owner_country	string	Country/Region
owner_phone_mobile	string	Mobile phone
owner_title	string	Title
owner_first_name	string	First name
owner_manager	string	Manager name
owner_address	string	Address
owner_company	string	Company
owner_last_name	string	Last name
owner_manager_email	string	Manager e-mail address

Attribute	Type	Description
owner_city	string	City
owner_department	string	Department
owner_load_ldap	button	Look up LDAP e-mail address
owner_manager_phone	string	Manager phone
owner_state	string	State
owner_office	string	Office
owner_zip	string	Zip code
owner_email	string	E-Mail
request_notes	string	Notes

Custom Attributes

Attribute	Type	Description	Values	Default
dns_servers	string	DNS server list		
sysprep_organization	string	Organization		
sysprep_password	string	New Administrator Password		
sysprep_custom_spec	string	Name		
sysprep_server_license_mod e	string	Identification	perServer, perSeat	
ldap_ous	string	LDAP Group		
sysprep_timezone	string	Timezone		
dns_suffixes	string	DNS Suffix list		
sysprep_product_id	string	ProductID		
sysprep_identification	string	Identification	domain, workgroup	
sysprep_per_server_max_co nnections	string	Maximum Connections		5
sysprep_computer_name	string	Computer Name		
sysprep_workgroup_name	string	Workgroup Name		WORKGROUP
sysprep_spec_override	boolean	Override Specification		false
addr_mode	string	Address Mode	static, dhcp	dhcp
linux_host_name	string	Computer Name		
sysprep_domain_admin	string	Domain Admin		
sysprep_change_sid	boolean	Change SID		true
sysprep_domain_name	string	Domain Name		
sysprep_upload_file	string	Upload		
gateway	string	Gateway		
ip_addr	string	IP Address		
linux_domain_name	string	Domain Name		
sysprep_domain_password	string	Domain Password		
sysprep_auto_logon	boolean	Auto Logon		true
sysprep_enabled	string	Customize		disabled
sysprep_delete_accounts	boolean	Delete Accounts		false
sysprep_upload_text	string	Sysprep Text		
wins_servers	string	WINS Server list		

Attribute	Type	Description	Values	Default
subnet_mask	string	Subnet Mask		
sysprep_full_name	string	Full Name		
sysprep_auto_logon_count	integer	Auto Logon Count	1, 2, 3	1
customization_template_id	integer	Script Name		
root_password	string	Root Password		
hostname	string	Host Name		
customization_template_script	string	Script Text		

Service Catalog Attributes

Attribute	Type	Description	Default
number_of_vms	integer	Count. <i>Max</i> : 50	1
vm_description	string	VM Description. <i>Max length</i> : 100	
vm_prefix	string	VM Name Prefix/Suffix	
src_vm_id	integer	Name	
vm_name	string	VM Name	
pxe_image_id	string	Image	
pxe_server_id	integer	Server	
host_name	string	Host Name	
provision_type	string	Provision Type	vmware
linked_clone	boolean	Linked Clone	false
snapshot	string	Snapshot	
vm_filter	integer	Filter	

Schedule Attributes

Attribute	Type	Description	Values	Default
schedule_type	string	When to Provision	schedule, immediately (<i>On Approval</i>)	immediately
vm_auto_start	boolean	Power on virtual machines after creation		true
schedule_time	time	Time to Provision on		
retirement	integer	Time until Retirement	0 (<i>Indefinite</i>), 1.month, 3.months, 6.months	0
retirement_warning	integer	Retirement Warning	1.week, 2.weeks, 30.days	1.week

Network Attributes

Attribute	Type	Description
vlan	string	vLan
mac_address	string	MAC Address

Hardware Attributes

Attribute	Type	Description	Values	Default
disk_format	string	Disk Format	thick, thin, unchanged	unchanged

Attribute	Type	Description	Values	Default
cpu_limit	integer	CPU (MHz)		
memory_limit	integer	Memory (MB)		
number_of_sockets	integer	Number of Sockets	1, 2, 4, 8	1
cores_per_socket	integer	Cores per Socket	1, 2, 4, 8	1
cpu_reserve	integer	CPU (MHz)		
vm_memory	string	Memory (MB)	1024, 2048, 4096	1024
memory_reserve	integer	Memory (MB)		
network_adapters	integer	Network Adapters	1, 2, 3, 4	1

Environment Attributes

These attributes cannot be passed directly. To use these attributes, provide them in the **additional_values** attribute group and have customization methods use them and modify the request accordingly.

Attribute	Type	Description	Values	Default
new_datastore_grow_increment	integer	Grow Increment (GB)		
new_datastore_create	boolean	Create Datastore		false
placement_cluster_name	integer	Name		
new_datastore_aggregate	string	Aggregate		
new_datastore_max_size	integer	Max Size (GB)		
new_datastore_storage_controller	string	Controller		
cluster_filter	integer	Filter		
host_filter	integer	Filter		
ds_filter	integer	Filter		
new_datastore_volume	string	Volume		
placement_host_name	integer	Name		
placement_ds_name	integer	Name		
new_datastore_fs_type	string	FS Type	NFS, VMFS	NFS
rp_filter	integer	Filter		
new_datastore_thin_provision	string	Thin Provision		
placement_auto	boolean	Choose Automatically		false
new_datastore_size	integer	Size (GB)		
new_datastore_autogrow	string	Autogrow		false
placement_folder_name	integer	Name		
new_datastore_name	string	Name		
placement_rp_name	integer	Name		
placement_dc_name	integer	Name		

- [Advanced Provisioning Attributes](#)
 - [Defining new vLans adapters](#)
 - [Defining new SCSI Controller](#)
 - [Defining new Disks](#)
 - [Cloning Type](#)
 - [Limit Template Selection](#)

Advanced Provisioning Attributes

This appendix lists parameters that apply to REST API initiated provisioning via /api/provision_requests.

- [Defining new vLans adapters](#)
- [Defining new SCSI Controller](#)
- [Defining new Disks](#)
- [Cloning Type](#)
- [Limit Template Selection](#)

Defining new vLans adapters

These attributes allow adding new vLan adapters during the provisioning request.

Attribute	Type	Required	Values	Default
network##	string	Required	Network Name	
network##.mac_address	string	Optional	MAC Address	
network##.deviceType	string	Optional	VirtualE1000, VirtualPCNet32, VirtualVmxnet	VirtualPCNet32
network##.connectable.allowGuestControl	boolean	Optional		true
network##.connectable.startConnected	boolean	Optional		true
network##.connectable.connected	boolean	Optional		true

Example:

```
"network0" : "VM Network",  
"network0.mac_address" : "00:50:56:57:58:59"
```

Defining new SCSI Controller

The SCSI Controller attributes are optional. If a controller is required, it will be added with the defined defaults. These settings are only required if you need to override a default value.

Attribute	Type	Required	Values	Default
ctrlscsi##	string	Required	VirtualBusLogicController, VirtualLsiLogicController	VirtualLsiLogicController
ctrlscsi##.sharedBus	string	Optional	noSharing, physicalSharing, virtualSharing	noSharing

Note:

The ## value defines the controller bus number

Example:

```
"ctrlscsi00" : "VirtualBusLogicController"  
"ctrlscsi00.sharedBus" : "virtualSharing"
```

Defining new Disks

The Disk attributes allow adding disks as part of the provisioning request.

Attribute	Type	Required	Values	Default
diskscsi##	string	Required	"controller_#:controller_position_#:size_in_mb"	
diskscsi##.datastore	string	Optional	Datastore Name	
diskscsi##.filename	string	Optional	Filename	
diskscsi##.connectable.allowGuestControl	boolean	Optional		false
diskscsi##.connectable.startConnected	boolean	Optional		true
diskscsi##.connectable.connected	boolean	Optional		true
diskscsi##.backing.diskmode	string	Optional	persistent, independent_persistent, independent_nonpersistent, nonpersistent	persistent
diskscsi##.backing.split	boolean	Optional		false
diskscsi##.backing.thinProvisioned	boolean	Optional		false
diskscsi##.backing.writeThrough	boolean	Optional		false

Example:

```
"diskscsi0" : "0:0:10"
"diskscsi1" : "0:0:8"
"diskscsi1.datastore" : "storage1"
"diskscsi1.backing.diskmode" : "independent_persistent"
```

Cloning Type

This attribute sets the type of cloning request, either clone to vm or clone to template. It is an attribute to set with the *template_fields* hash.

Attribute	Type	Required	Values	Default
request_type	string	Optional	clone_to_vm, clone_to_template	clone_to_vm

Example:

```
"template_fields" : {
  "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742",
  "request_type" : "clone_to_template"
}
```

Limit Template Selection

This attribute value limits the template selection to the Datacenters contained in the specified cluster. It is an attribute to be specified with the *additional_values* hash.

Attribute	Type	Required
cluster	string	Optional

Example:

```
"additional_values" : {
  "request_id" : "87458",
  "cluster" : "sales_prod"
}
```

- [SOAP to REST Mapping guide](#)
 - [Insight Web Services](#)
 - [Control Web Services](#)
 - [Automate Web Services](#)

SOAP to REST Mapping guide

- [Insight Web Services](#)
- [Control Web Services](#)
- [Automate Web Services](#)

Insight Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
ClusterGetTags	GET	/api/cluster/:id/tags		
DatastoreGetTags	GET	/api/data_stores/:id/tags		
EmsGetTags	GET	/api/providers/:id/tags		
EVMDeleteVmByName	DELETE	/api/vms/:id		Delete VM
	POST	/api/vms/:id	delete	Delete VMs
EVMClusterList	GET	/api/clusters		
EVMDatastoreList	GET	/api/data_stores		
EVMHostList	GET	/api/hosts		
EVMResourcePoolList	GET	/api/resource_pools		
EVMPing	GET	/api		
EVMVmAccounts	GET	/api/vms/:id/accounts		
EVMVmList	GET	/api/vms		
	GET	/api/hosts/:id?attributes=vms		
	GET	/api/hosts?filter[]=guid='...&expand=resources&attributes=vms		
EVMVmSoftware	GET	/api/vms:id/software		
FindClusterById	GET	/api/clusters/:id		
FindClustersById	GET	/api/clusters/:id		
FindDatastoreById	GET	/api/data_stores/:id		
FindDatastoresById	GET	/api/data_stores/:id		

SOAP API	HTTP Method	REST API URLs	Actions	Examples
FindEmsByGuid	GET	/api/providers?filter[]=guid='....'		
FindHostByGuid	GET	/api/hosts?filter[]='guid='....'"		
FindHostsByGuid	GET	/api/hosts?filter[]='guid='....'"		
FindResourcePoolById	GET	/api/resource_pools/:id		
FindResourcePoolsById	GET	/api/resource_pools		
	GET	/api/resource_pools?filter[]=id=:id\&filter[]=or id=:id&...		
FindVmByGuid	GET	/api/vms?filter[]=guid='....'		
FindVmsByGuid	GET	/api/vms?filter[]=guid='...'\&filter[]=or guid='....'\&...		
GetClusterList	GET	/api/clusters		
	GET	/api/clusters?filter[]=ems_id=:id		
	GET	/api/providers/:id?attributes=ems_clusters		
GetClustersByList	GET	/api/clusters/:id		
	GET	/api/clusters?filter[]=id=:id\&filter[]=or id=:id&...		
GetClustersByTag	GET	/api/clusters?by_tag=:tag_path		
GetDatastoreByList	GET	/api/data_stores		
GetDatastoreByTag	GET	/api/data_stores?by_tag=:tag_path		
GetEmsByList	GET	/api/providers		
	GET	/api/providers/:id		
	GET	/api/providers?filter[]=...'\&expand=resources		
GetEmsList	GET	/api/providers		
GetHostsByList	GET	/api/hosts		
	GET	/api/hosts?filter[]=...'\&expand=resources		
GetHostList	GET	/api/hosts		
	GET	/api/providers/:id?attributes=hosts		
	GET	/api/providers?filter[]=guid='...'\&expand=resources\&attributes=hosts		
GetResourcePoolList	GET	/api/providers/:id?attributes=resource_pools		
	GET	/api/providers?filter[]=guid='...'\&expand=resources\&attributes=resource_pools		
GetResourcePoolsByList	GET	/api/resource_pools		

SOAP API	HTTP Method	REST API URLs	Actions	Examples
	GET	/api/resource_pools/:id		
	GET	/api/resource_pools?filter[]=... .\&expand=resources		
GetResourcePoolsByTag	GET	/api/resource_pools? by_tag=:tag_path		
GetTemplatesByTag	GET	/api/templates?by_path=:tag_path		
GetVmList	GET	/api/hosts/:id?attributes=vms		
	GET	/api/hosts?filter[]=guid='... '.\&expand=resources\&attributes=vms		
GetVmsByList	GET	/api/vms		
	GET	/api/vms/:id		
	GET	/api/vms?filter[]=... .\&expand=resources		
GetVmsByTag	GET	/api/vms?by_path=:tag_path		
HostGetTags	GET	/api/hosts/:id/tags		
	GET	/api/hosts/:id?expand=tags		
	GET	/api/hosts?filter[]=guid='... '.\&expand=resources,tags		
ResourcePoolGetTags	GET	/api/resource_pools/:id/tags		
	GET	/api/resource_pools/:id? expand=tags		
VmGetTags	GET	/api/vms/:id/tags		
	GET	/api/vms/:id?expand=tags		

Control Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
ClusterSetTag	POST	/api/clusters/:id/tags	assign	Assigning Tags
DatastoreSetTag	POST	/api/data_stores/:id/tags	assign	Assigning Tags
EmsSetTag	POST	/api/providers/:id/tags	assign	Assigning Tags
EVMActionList	GET	/api/policies/:id/policy_actions		Policy Actions
	GET	/api/policies/:id? expand=policy_actions		
	GET	/api/policies?filter[]=guid='... '.\&expand=resources,policy_actions		
EVMAssignPolicy	POST	/api/hosts/:id/policies	assign	Assigning Policies
EVMConditionList	GET	/api/policies/:id/conditions		Policy Conditions
	GET	/api/policies/:id? expand=conditions		

SOAP API	HTTP Method	REST API URLs	Actions	Examples
	GET	/api/policies?filter[]=guid='...'\&expand=resources,conditions		
EVMEventList	GET	/api/policies/:id/events		Policy Events
	GET	/api/policies/:id?expand=events		
	GET	/api/policies?filter[]=guid='...'\&expand=resources,events		
EVMGetPolicy	GET	/api/policies		Querying Policies
	GET	/api/policies/:id		
	GET	/api/policies?filter[]=name='....'		
EVMPolicyList	GET	/api/hosts/:id/policies		Policy Management
	GET	/api/hosts/:id?expand=policies		
	GET	/api/hosts?filter[]=guid='...'\&expand=resources,policies		
EVMVmEventByProperty	POST	/api/vms/:id	add_event	Add VM Event
EVMVmScanByProperty	POST	/api/vms/:id	scan	Scan VM
EVMSmartStart	POST	/api/vms/:id	start	Start VM
EVMSmartStop	POST	/api/vms/:id	stop	Stop VM
EVMSmartSuspend	POST	/api/vms/:id	suspend	Suspend VM
EVMUnassignPolicy	POST	/api/hosts/:id/policies	unassign	Unassigning Policies
EVMVmRsop	POST	/api/vms/:id/policies/:id	resolve	Resolving Policies
	POST	/api/vms/:id/policy_profiles/:id	resolve	Resolving Policy Profiles
HostSetTag	POST	/api/hosts/:id/tags	assign	Assigning Tags
ResourcePoolSetTag	POST	/api/resource_pools/:id/tags	assign	Assigning Tags
VmAddCustomAttribute	POST	/api/vms/:id/custom_attributes	add	Adding Custom Attributes
VmAddCustomAttributes	POST	/api/vms/:id/custom_attributes	add	
VmAddCustomAttributeByFields	POST	/api/vms/:id/custom_attributes	add	
VmDeleteCustomAttribute	POST	/api/vms/:id/custom_attributes	delete	Deleting Custom Attributes
VmDeleteCustomAttributes	POST	/api/vms/:id/custom_attributes	delete	
VmSetTag	POST	/api/vms/:id/tags	assign	Assigning Tags
VmSetOwner	POST	/api/vms/:id	set_owner	Set Owner of VM

Automate Web Services

SOAP API	HTTP Method	REST API URLs	Actions	Examples
CreateAutomationRequest	POST	/api/automation_requests	create	Create Automation Request , Create Automation Requests
EVMProvisionRequestEx	POST	/api/provision_requests	create	Create Provision Request , Create Provision Requests
GetAutomationRequest	GET	/api/automation_requests/:id		
GetAutomationTask	GET	/api/automation_requests/:id?expand=request_tasks		
	GET	/api/automation_requests/:id/request_tasks/:id		
GetVmProvisionRequest	GET	/api/provision_requests/:id		
	GET	/api/provision_requests/:id?expand=request_tasks		
VmProvisionRequest	POST	/api/provision_requests	create	Create Provision Request , Create Provision Requests

- [Deprecated Types](#)
 - [API v2.4.0 ⇒ v3.0.0](#)

Deprecated Types

This page lists the types that have been deprecated in previous releases and have been updated to newer types. They are *temporarily* supported in the listed releases but will be removed in future releases.

API v2.4.0 ⇒ v3.0.0

For Orchestration Templates:

Old Type	New Type
OrchestrationTemplateAzure	ManageIQ::Providers::Azure::CloudManager::OrchestrationTemplate
OrchestrationTemplateHot	ManageIQ::Providers::Openstack::CloudManager::OrchestrationTemplate
OrchestrationTemplateCfn	ManageIQ::Providers::Amazon::CloudManager::OrchestrationTemplate
OrchestrationTemplateVnfd	ManageIQ::Providers::Openstack::CloudManager::VnfdTemplate

- [Deprecated Resource Attributes](#)
 - [API v2.2.0 ⇒ v2.3.0](#)

Deprecated Resource Attributes

This page lists the resource attributes that have been deprecated in earlier releases and have been either renamed to newer attribute names or simply removed.

API v2.2.0 ⇒ v2.3.0

For Hardware:

Old Attribute Name	New Attribute Name
numvcpus	cpu_sockets
memory_cpu	memory_mb

For Metrics:

Old Attribute Name	New Attribute Name
v_derived_logical_cpus_used	v_derived_cpu_total_cores_used

For Provision Workflows:

Old Attribute Name	New Attribute Name
allowed_datastore_aggregate	removed
allowed_datastore_storage_controller	removed

For all:

Old Attribute Name	New Attribute Name
cores_per_socket	cpu_cores_per_socket
logical_cpus	cpu_total_cores
aggregate_logical_cpus	aggregate_cpu_total_cores

Infrastructure Management API Examples

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Foundation](#)
- [Automation Requests](#)
- [Providers](#)
- [Provisioning Requests](#)
- [Service Catalogs](#)
- [Service Templates](#)
- [Services](#)
- [VM Management](#)
- [Tag Management](#)
- [Usage Flows](#)

Infrastructure Management API Examples - Foundation

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Querying Examples](#)
- [Paging Queries](#)
- [Example Queries:](#)
 - [Query All VMs](#)
 - [Query A Specific VM](#)
 - [Query all VMs, but return only return name and vendor](#)
 - [Query VMs named sample* and return name and vendor](#)

- [Query VMs but only return the first 500](#)
- [Query VMs but return the second 500](#)
- [Query first 1000 VMs, named test*, get name, vendor and guid and sort by name in ascending order](#)
- [Query Services tagged for the finance department](#)
- [Query Services tagged with the finance or engineering department](#)
- [Get details on the Tags of the first service](#)
- [Get details of the first service catalog and related details on the assigned service templates](#)
- [Get services requests created from a specific template and get details on all related request tasks.](#)
- [Get a specific provision request with expanded details on the associated provision request tasks](#)

Example Queries:

Query All VMs

```
GET /api/vms
```

Query A Specific VM

```
GET /api/vms/1386
```

Query all VMs, but return only return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor
```

Query VMs named sample* and return name and vendor

```
GET /api/vms?expand=resources&attributes=name,vendor&filter[]=name='sample%'
```

Query VMs but only return the first 500

```
GET /api/vms?offset=0&limit=500
```

Query VMs but return the second 500

```
GET /api/vms?offset=500&limit=500
```

Query first 1000 VMs, named test*, get name, vendor and guid and sort by name in ascending order

```
GET /api/vms?offset=0&limit=1000&filter[]=name='test%'
    &expand=resources&attributes=name,vendor,guid
    &sort_by=name&sort_order=asc
```

Query Services tagged for the finance department

```
GET /api/services?by_tag=/department/finance
```

Query Services tagged with the finance or engineering department

```
GET /api/services?by_tag=/department/finance,/department/engineering
```

Get details on the Tags of the first service

```
GET /api/services/1/tags?expand=resources
```

Get details of the first service catalog and related details on the assigned service templates

```
GET /api/service_catalogs/1?expand=service_templates
```

Get services requests created from a specific template and get details on all related request tasks.

```
GET /api/service_templates/25/service_requests?expand=resources,request_tasks
```

Get a specific provision request with expanded details on the associated provision request tasks

```
GET /api/provision_requests/120?expand=request_tasks
```

- [Paging Queries](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)

Paging Queries

In this series of requests, we will show the paging queries and the expected responses for each subsequent page.

Request:

```
GET /api/vms?offset=0&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:

```
{
  "name": "vms",
  "count": 1912,
  "subcount": 500,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/176",
      "id": "176",
      "name": "53 Zone1"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/1575",
      "id": "1575",
      "name": "VmEmpty-3de98f0f-c6f3-4f8b-a932-554713a61067"
    }
  ],
}
```

```
"actions": [  
]  
}
```

Request:

```
GET /api/vms?offset=500&limit=500  
    &sort_order=asc&sort_by=name  
    &expand=resources&attributes=name
```

Response:

```
{  
  "name": "vms",  
  "count": 1912,  
  "subcount": 500,  
  "resources": [  
    {  
      "href": "http://localhost:3000/api/vms/1574",  
      "id": "1574",  
      "name": "VmEmpty-3e13ff43-6907-4a22-8f95-58aeb1bffa0b"  
    },  
    ...  
    {  
      "href": "http://localhost:3000/api/vms/1076",  
      "id": "1076",  
      "name": "VmEmpty-9a885181-7771-4f91-9805-245c7606d833"  
    }  
  ],  
  "actions": [  
  ]  
}
```

Request:

```
GET /api/vms?offset=1000&limit=500  
    &sort_order=asc&sort_by=name  
    &expand=resources&attributes=name
```

Response:

```
{  
  "name": "vms",  
  "count": 1912,  
  "subcount": 500,  
  "resources": [  
    {  
      "href": "http://localhost:3000/api/vms/1074",  
      "id": "1074",  
      "name": "VmEmpty-9ab9e101-92b0-4b6b-864e-e196538da8a8"  
    },  
    ...  
    {  
      "href": "http://localhost:3000/api/vms/575",  
      "id": "575",  
      "name": "VmEmpty-f251f135-01c8-4d44-b8e1-37b30844a9dd"  
    }  
  ],  
  "actions": [  
  ]  
}
```


Request:

```
GET /api/vms?offset=1500&limit=500
    &sort_order=asc&sort_by=name
    &expand=resources&attributes=name
```

Response:

```
{
  "name": "vms",
  "count": 1912,
  "subcount": 412,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/574",
      "id": "574",
      "name": "VmEmpty-f28912f3-b096-487f-9763-97b39b67364b"
    },
    ...
    {
      "href": "http://localhost:3000/api/vms/1907",
      "id": "1907",
      "name": "yy_vm"
    }
  ],
  "actions": [
  ]
}
```

Note:

In this last request, the **subcount** was less than the requested page size thus denoting the last page of data being returned.

Infrastructure Management API Examples - Automation Requests

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Create an Automation Request](#)
- [Create multiple Automation Requests](#)
- [Trigger a single Automation Request](#)
 - [Request:](#)
 - [Request:](#)
 - [Response:](#)

Trigger a single Automation Request

In the automation requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

POST /api/automation_requests

```
{
  "version" : "1.1",
  "uri_parts" : {
    "namespace" : "System",
    "class" : "Request",
    "instance" : "InspectME",
    "message" : "create"
  },
  "parameters" : {
    "var1" : "xxxxx",
    "var2" : "YYYYY",
    "var3" : 1024,
    "var4" : true,
    "var5" : "last value"
  },
  "requester" : {
    "user_name" : "admin",
    "auto_approve" : true
  }
}
```

Optionally, the action based request format is also supported:

Request:

POST /api/automation_requests

```
{
  "action" : "create",
  "resource" : {
    "version" : "1.1",
    "uri_parts" : {
      "namespace" : "System",
      "class" : "Request",
      "instance" : "InspectME",
      "message" : "create"
    },
    "parameters" : {
      "var1" : "xxxxx",
      "var2" : "YYYYY",
      "var3" : 1024,
      "var4" : true,
      "var5" : "last value"
    },
    "requester" : {
      "user_name" : "admin",
      "auto_approve" : true
    }
  }
}
```

Response:

```
{
  "results": [
    {
      "id": "12",
      "href" : "http://localhost:3000/api/automation_requests/12",

```

```

    "description": "Automation Task",
    "approval_state": "approved",
    "type": "AutomationRequest",
    "created_on": "2015-04-16T21:49:55Z",
    "updated_on": "2015-04-16T21:49:55Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "automation",
    "request_state": "pending",
    "status": "Ok",
    "options": {
      "message": "create",
      "namespace": "System",
      "class_name": "Request",
      "instance_name": "InspectME",
      "user_id": "1",
      "attrs": {
        "var1": "xxxxxx",
        "var2": "yyyyyy",
        "var3": 1024,
        "var4": true,
        "var5": "last value",
        "userid": "admin"
      }
    },
    "userid": "admin"
  }
]
}

```

- [Trigger multiple Automation Requests](#)
 - [Request:](#)
 - [Response:](#)

Trigger multiple Automation Requests

In the automation requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

POST /api/automation_requests

```

{
  "action" : "create",
  "resources" : [
    {
      "version" : "1.1",
      "uri_parts" : {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
      },
      "parameters" : {
        "vm_name" : "test_1",
        "var2" : "yyyyyy",
        "var3" : 1024,
        "var4" : true,

```

```

        "var5" : "last value"
    },
    "requester" : {
        "user_name" : "jdoe",
        "auto_approve" : true
    }
},
{
    "uri_parts" : {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
    },
    "parameters" : {
        "vm_name" : "test_2",
        "vm_memory" : 1024,
        "memory_limit" : 16384
    },
    "requester" : {
        "auto_approve" : true
    }
},
{
    "uri_parts" : {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
    },
    "parameters" : {
        "vm_name" : "test_3",
        "vm_memory" : 2048,
        "memory_limit" : 16384
    },
    "requester" : {
        "auto_approve" : true
    }
},
{
    "uri_parts" : {
        "namespace" : "System",
        "class" : "Request",
        "instance" : "InspectME",
        "message" : "create"
    },
    "parameters" : {
        "vm_name" : "test_4",
        "vm_memory" : 4096,
        "memory_limit" : 16384
    },
    "requester" : {
        "auto_approve" : true
    }
}
]
}

```

Response:

```

{
  "results": [
    {

```

```

    "id": "14",
    "href" : "http://localhost:3000/api/automation_requests/14",
    "description": "Automation Task",
    "approval_state": "approved",
    "type": "AutomationRequest",
    "created_on": "2015-04-16T21:59:42Z",
    "updated_on": "2015-04-16T21:59:42Z",
    "requester_id": "13",
    "requester_name": "aab",
    "request_type": "automation",
    "request_state": "pending",
    "status": "Ok",
    "options": {
      "message": "create",
      "namespace": "System",
      "class_name": "Request",
      "instance_name": "InspectME",
      "user_id": "13",
      "attrs": {
        "vm_name": "test_1",
        "var2": "YYYYY",
        "var3": 1024,
        "var4": true,
        "var5": "last value",
        "userid": "aab"
      }
    },
    "userid": "aab"
  },
  {
    "id": "15",
    "href" : "http://localhost:3000/api/automation_requests/15",
    "description": "Automation Task",
    "approval_state": "approved",
    "type": "AutomationRequest",
    "created_on": "2015-04-16T21:59:42Z",
    "updated_on": "2015-04-16T21:59:42Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "automation",
    "request_state": "pending",
    "status": "Ok",
    "options": {
      "message": "create",
      "namespace": "System",
      "class_name": "Request",
      "instance_name": "InspectME",
      "user_id": "1",
      "attrs": {
        "vm_name": "test_2",
        "vm_memory": 1024,
        "memory_limit": 16384,
        "userid": "admin"
      }
    },
    "userid": "admin"
  },
  {
    "id": "16",
    "href" : "http://localhost:3000/api/automation_requests/16",
    "description": "Automation Task",
    "approval_state": "approved",
    "type": "AutomationRequest",
    "created_on": "2015-04-16T21:59:42Z",

```

```

    "updated_on": "2015-04-16T21:59:42Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "automation",
    "request_state": "pending",
    "status": "Ok",
    "options": {
      "message": "create",
      "namespace": "System",
      "class_name": "Request",
      "instance_name": "InspectME",
      "user_id": "1",
      "attrs": {
        "vm_name": "test_3",
        "vm_memory": 2048,
        "memory_limit": 16384,
        "userid": "admin"
      }
    },
    "userid": "admin"
  },
  {
    "id": "17",
    "href" : "http://localhost:3000/api/automation_requests/17",
    "description": "Automation Task",
    "approval_state": "approved",
    "type": "AutomationRequest",
    "created_on": "2015-04-16T21:59:42Z",
    "updated_on": "2015-04-16T21:59:42Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "automation",
    "request_state": "pending",
    "status": "Ok",
    "options": {
      "message": "create",
      "namespace": "System",
      "class_name": "Request",
      "instance_name": "InspectME",
      "user_id": "1",
      "attrs": {
        "vm_name": "test_4",
        "vm_memory": 4096,
        "memory_limit": 16384,
        "userid": "admin"
      }
    },
    "userid": "admin"
  }
]
}

```

Infrastructure Management API Examples - Providers

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Create a Provider](#)
- [Create a Provider with multiple credentials](#)
- [Update a Provider](#)

- [Refresh a Provider](#)
- [Delete a Provider](#)
- [Create Provider](#)
 - [Request:](#)
 - [Response:](#)

Create Provider

Request:

POST /api/providers

```
{
  "type"      : "ManageIQ::Providers::Redhat::InfraManager",
  "name"      : "rhevm101",
  "hostname"  : "rhevm101",
  "ipaddress" : "100.200.300.101",
  "credentials" : {
    "userid"   : "admin_account",
    "password" : "admin_password"
  }
}
```

Response:

```
{
  "results": [
    {
      "id": "105",
      "href" : "http://localhost:3000/api/providers/105",
      "name": "rhevm101",
      "hostname": "rhevm101",
      "ipaddress": "100.200.300.101",
      "created_on": "2015-05-05T15:47:41Z",
      "updated_on": "2015-05-05T15:47:41Z",
      "guid": "10360312-f33e-11e4-86c7-b8e85646e742",
      "zone_id": "1",
      "type": "ManageIQ::Providers::Redhat::InfraManager"
    }
  ]
}
```

- [Create Provider](#)
 - [Request:](#)
 - [Response:](#)

Create Provider

Request:

POST /api/providers

```
{
  "type"      : "ManageIQ::Providers::Redhat::InfraManager",
  "name"      : "rhevm102",
  "hostname"  : "rhevm102",
  "ipaddress" : "100.200.300.102",
  "credentials" : [
```

```
{
  {
    "userid" : "default_userid",
    "password" : "default_password"
  },
  {
    "userid" : "metrics_userid",
    "password" : "metrics_password",
    "auth_type" : "metrics"
  }
}
]
```

Response:

```
{
  "results": [
    {
      "id": "106",
      "href" : "http://localhost:3000/api/providers/106",
      "name": "rhevm102",
      "hostname": "rhevm102",
      "ipaddress": "100.200.300.102",
      "created_on": "2015-05-05T15:57:44Z",
      "updated_on": "2015-05-05T15:57:44Z",
      "guid": "acbd610e-f3f6-11e4-aaba-b8e85646e742",
      "zone_id": "1",
      "type": "ManageIQ::Providers::Redhat::InfraManager"
    }
  ]
}
```

- [Update Provider](#)
 - [Request:](#)
 - [Response:](#)

Update Provider

Request:

POST /api/providers/106

```
{
  "action" : "edit",
  "ipaddress" : "100.200.300.112",
  "credentials" : [
    {
      "userid" : "updated_metrics_userid",
      "password" : "updated_metrics_password",
      "auth_type" : "metrics"
    }
  ]
}
```

Response:

```
{
  "href": "http://localhost:3000/api/providers/106",
  "id": "106",
  "name": "rhevm102",
  "hostname": "rhevm102",
  "ipaddress": "100.200.300.112",
```



```

"created_on": "2015-05-06T13:49:11Z",
"updated_on": "2015-05-06T13:53:06Z",
"guid": "acbd610e-f3f6-11e4-aaba-b8e85646e742",
"zone_id": "1",
"type": "ManageIQ::Providers::Redhat::InfraManager"
}

```

- [Refresh Provider](#)
 - [Request:](#)
 - [Response:](#)

Refresh Provider

Request:

```
POST /api/providers/105
```

```

{
  "action" : "refresh"
}

```

Response:

```

{
  "success": true,
  "message": "Provider id:105 name:'sample_provider' refreshing",
  "href": "http://localhost:3000/api/providers/105",
  "task_id" : "23",
  "task_href" : "http://localhost:3000/api/tasks/23",
  "tasks" : [
    { "id" : "23", "href" : "http://localhost:3000/api/tasks/23"},
    { "id" : "24", "href" : "http://localhost:3000/api/tasks/24"}
  ]
}

```

Note: In the above example a multi-manager provider was refreshed and in the response, a task id and href pair is returned for each of the provider managers. This allows the caller to check the status of each manager refresh for completion.

- [Delete Provider](#)
 - [Deleting a single Provider](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)
 - [Deleting multiple Providers](#)
 - [Request:](#)
 - [Response:](#)

Delete Provider

Deleting a single Provider

Request:

```
DELETE /api/provider/105
```

Or via the delete action as follows:

```
POST /api/provider/105
```

```
{
  "action" : "delete"
}
```

Response:

```
{
  "success": true,
  "message": "Provider id:106 name:'rhevm102' deleting",
  "task_id": "625",
  "task_href": "http://localhost:3000/api/tasks/625",
  "href": "http://localhost:3000/api/providers/106"
}
```

Note:

Delete actions are done asynchronously as it could take a while to complete. The delete task can be queried if needed as follows

Request:

```
GET /api/tasks/625
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/625",
  "id": "625",
  "name": "Provider id:106 name:'rhevm102' deleting",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-06T14:02:26Z",
  "updated_on": "2015-05-06T14:02:32Z"
}
```

Deleting multiple Providers

Request:

```
POST /api/providers
```

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/providers/107" },
    { "href" : "http://localhost:3000/api/providers/108" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,

```

```

    "message": "Provider id:107 name:'rhevm102' deleting",
    "task_id": "626",
    "task_href": "http://localhost:3000/api/tasks/626",
    "href": "http://localhost:3000/api/providers/107"
  },
  {
    "success": true,
    "message": "Provider id:108 name:'rhevm103' deleting",
    "task_id": "627",
    "task_href": "http://localhost:3000/api/tasks/627",
    "href": "http://localhost:3000/api/providers/108"
  }
]
}

```

Infrastructure Management API Examples - Provisioning Requests

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Create a Provision Request](#)
- [Create multiple Provision Requests](#)
- [Trigger a single Provision Request](#)
 - [Request:](#)
 - [Response:](#)

Trigger a single Provision Request

In the provision requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

POST /api/provision_requests

```

{
  "version" : "1.1",
  "template_fields" : {
    "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
  },
  "vm_fields" : {
    "number_of_cpus" : 1,
    "vm_name" : "aab_rest_vm1",
    "vm_memory" : "1024",
    "vlan" : "rhevm"
  },
  "requester" : {
    "user_name" : "jdoe",
    "owner_first_name" : "John",
    "owner_last_name" : "Doe",
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  }
}

```

```

    },
    "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
    },
    "additional_values" : {
        "request_id" : "1001"
    },
    "ems_custom_attributes" : { },
    "miq_custom_attributes" : { }
}

```

Optionally, the action based request format is also supported:

POST /api/provision_requests

```

{
    "action" : "create",
    "resource" : {
        "version" : "1.1",
        "template_fields" : {
            "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742"
        },
        "vm_fields" : {
            "number_of_cpus" : 1,
            "vm_name" : "aab_rest_vm1",
            "vm_memory" : "1024",
            "vlan" : "rhevm"
        },
        "requester" : {
            "user_name" : "jdoe",
            "owner_first_name" : "John",
            "owner_last_name" : "Doe",
            "owner_email" : "jdoe@sample.com",
            "auto_approve" : true
        },
        "tags" : {
            "network_location" : "Internal",
            "cc" : "001"
        },
        "additional_values" : {
            "request_id" : "1001"
        },
        "ems_custom_attributes" : { },
        "miq_custom_attributes" : { }
    }
}

```

Response:

```

{
    "results": [
        {
            "href" : "http://localhost:3000/api/provision_requests/18",
            "id": "18",
            "description": "Provision from [bd-clone-template] to [aab_rest_vm1]",
            "approval_state": "pending_approval",
            "type": "MiqProvisionRequest",
            "created_on": "2015-05-08T17:42:55Z",
            "updated_on": "2015-05-08T17:42:55Z",
            "requester_id": "1",
            "requester_name": "Administrator",
            "request_type": "template",

```

```

"request_state": "pending",
"message": "VM Provisioning - Request Created",
"status": "Ok",
"options": {
  "use_pre_dialog": false,
  "request_type": "template",
  "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
  "owner_first_name": "John",
  "owner_last_name": "Doe",
  "owner_email": "jdoe@sample.com",
  "vm_tags": [
    62,
    58
  ],
  "addr_mode": [
    "static",
    "Static"
  ],
  "placement_cluster_name": [
    null,
    null
  ],
  "cluster_filter": [
    null,
    null
  ],
  "placement_auto": [
    true,
    1
  ],
  "placement_dc_name": [
    null,
    null
  ],
  "number_of_vms": [
    1,
    "1"
  ],
  "src_vm_id": [
    1947,
    "bd-clone-template"
  ],
  "provision_type": [
    "native_clone",
    "Native Clone"
  ],
  "linked_clone": [
    null,
    null
  ],
  "vm_name": "aab_rest_vm1",
  "pxe_server_id": [
    null,
    null
  ],
  "schedule_type": [
    "immediately",
    "Immediately on Approval"
  ],
  "vm_auto_start": [
    true,
    1
  ],
  "schedule_time": "2015-05-09T13:42:54-04:00",

```

```

    "retirement": [
        0,
        "Indefinite"
    ],
    "retirement_warn": [
        604800,
        "1 Week"
    ],
    "stateless": [
        false,
        0
    ],
    "vlan": [
        "rhevm",
        "rhevm"
    ],
    "disk_format": [
        "default",
        "Default"
    ],
    "number_of_sockets": [
        1,
        "1"
    ],
    "cores_per_socket": [
        1,
        "1"
    ],
    "vm_memory": [
        "1024",
        "1024"
    ],
    "network_adapters": [
        1,
        "1"
    ],
    "placement_host_name": [
        null,
        null
    ],
    "placement_ds_name": [
        null,
        null
    ],
    "src_vm_nics": [

],
    "src_vm_lans": [

],
    "customize_enabled": [
        "enabled"
    ],
    "src_ems_id": [
        105,
        "rhevm230"
    ],
    "auto_approve": false,
    "ws_values": {
        "request_id": "1001"
    },
    "ws_ems_custom_attributes": {
    },
    "ws_miq_custom_attributes": {

```

```

    }
  },
  "userid": "jdoe",
  "source_id": 1947,
  "source_type": "VmOrTemplate"
}
]
}

```

- [Trigger multiple Provision Requests](#)
 - [Request:](#)
 - [Response:](#)

Trigger multiple Provision Requests

In the provision requests:

- version defaults to “1.1” if not specified.
- user_name defaults to the REST API authenticated user if not specified.

Request:

POST /api/provision_requests

```

{
  "action" : "create",
  "resources" : [
    {
      "version" : "1.1",
      "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
      "vm_fields" : {
        "vm_name" : "jdoe_rest_vm1",
        "number_of_cpus" : 1,
        "vm_memory" : "1024",
        "vlan" : "nic1"
      },
      "requester" : {
        "user_name" : "jdoe",
        "owner_first_name" : "John",
        "owner_last_name" : "Doe",
        "owner_email" : "jdoe@sample.com",
        "auto_approve" : true
      },
      "tags" : {
        "network_location" : "Internal",
        "cc" : "001"
      },
      "additional_values" : { "request_id" : "1001" },
      "ems_custom_attributes" : { },
      "miq_custom_attributes" : { }
    },
    {
      "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
      "vm_fields" : {
        "vm_name" : "jdoe_rest_vm2",
        "number_of_cpus" : 1,
        "vm_memory" : "2048",
        "vlan" : "nic1"
      },
      "requester" : {
        "owner_first_name" : "John",

```

```

    "owner_last_name" : "Doe",
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  },
  "tags" : {
    "network_location" : "Internal",
    "cc" : "001"
  },
  "additional_values" : { "request_id" : "1002" }
},
{
  "template_fields" : { "guid" : "afe6e8a0-89fd-11e3-b6ac-b8e85646e742" },
  "vm_fields" : {
    "vm_name" : "jdoe_rest_vm3",
    "number_of_cpus" : 1,
    "vm_memory" : "4096",
    "vlan" : "nic1"
  },
  "requester" : {
    "owner_first_name" : "John",
    "owner_last_name" : "Doe",
    "owner_email" : "jdoe@sample.com",
    "auto_approve" : true
  },
  "tags" : {
    "network_location" : "Internal",
    "cc" : "001"
  },
  "additional_values" : { "request_id" : "1003" }
}
]
}

```

Response:

```

{
  "results": [
    {
      "href": "http://localhost:3000/api/provision_requests/19",
      "id": "19",
      "description": "Provision from [bd-clone-template] to [aab_rest_vm1]",
      "approval_state": "pending_approval",
      "type": "MiqProvisionRequest",
      "created_on": "2015-05-08T18:25:25Z",
      "updated_on": "2015-05-08T18:25:26Z",
      "requester_id": "1",
      "requester_name": "jdoe",
      "request_type": "template",
      "request_state": "pending",
      "message": "VM Provisioning - Request Created",
      "status": "Ok",
      "options": {
        "use_pre_dialog": false,
        "request_type": "template",
        "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
        "owner_first_name": "John",
        "owner_last_name": "Doe",
        "owner_email": "jdoe@sample.com",
        "vm_tags": [
          62,
          58
        ],
        "addr_mode": [

```



```

        "static",
        "Static"
    ],
    "placement_cluster_name": [
        null,
        null
    ],
    "cluster_filter": [
        null,
        null
    ],
    "placement_auto": [
        true,
        1
    ],
    "placement_dc_name": [
        null,
        null
    ],
    "number_of_vms": [
        1,
        "1"
    ],
    "src_vm_id": [
        1947,
        "bd-clone-template"
    ],
    "provision_type": [
        "native_clone",
        "Native Clone"
    ],
    "linked_clone": [
        null,
        null
    ],
    "vm_name": "aab_rest_vm1",
    "pxe_server_id": [
        null,
        null
    ],
    "schedule_type": [
        "immediately",
        "Immediately on Approval"
    ],
    "vm_auto_start": [
        true,
        1
    ],
    "schedule_time": "2015-05-09T14:25:25-04:00",
    "retirement": [
        0,
        "Indefinite"
    ],
    "retirement_warn": [
        604800,
        "1 Week"
    ],
    "stateless": [
        false,
        0
    ],
    "vlan": [
        "rhevm",
        "rhevm"
    ]

```

```

    ],
    "disk_format": [
        "default",
        "Default"
    ],
    "number_of_sockets": [
        1,
        "1"
    ],
    "cores_per_socket": [
        1,
        "1"
    ],
    "vm_memory": [
        "1024",
        "1024"
    ],
    "network_adapters": [
        1,
        "1"
    ],
    "placement_host_name": [
        null,
        null
    ],
    "placement_ds_name": [
        null,
        null
    ],
    ],
    "src_vm_nics": [

],
    "src_vm_lans": [

],
    "customize_enabled": [
        "enabled"
    ],
    "src_ems_id": [
        105,
        "rhevm230"
    ],
    "auto_approve": false,
    "ws_values": {
        "request_id": "1001"
    },
    "ws_ems_custom_attributes": {
    },
    "ws_miq_custom_attributes": {
    }
},
    "userid": "jdoe",
    "source_id": "1947",
    "source_type": "VmOrTemplate"
},
{
    "href": "http://localhost:3000/api/provision_requests/20",
    "id": "20",
    "description": "Provision from [bd-clone-template] to [aab_rest_vm2]",
    "approval_state": "pending_approval",
    "type": "MiqProvisionRequest",
    "created_on": "2015-05-08T18:25:28Z",
    "updated_on": "2015-05-08T18:25:29Z",
    "requester_id": "1",

```

```

"requester_name": "jdoe",
"request_type": "template",
"request_state": "pending",
"message": "VM Provisioning - Request Created",
"status": "Ok",
"options": {
  "use_pre_dialog": false,
  "request_type": "template",
  "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
  "owner_first_name": "John",
  "owner_last_name": "Doe",
  "owner_email": "jdoe@sample.com",
  "vm_tags": [
    62,
    58
  ],
  "addr_mode": [
    "static",
    "Static"
  ],
  "placement_cluster_name": [
    null,
    null
  ],
  "cluster_filter": [
    null,
    null
  ],
  "placement_auto": [
    true,
    1
  ],
  "placement_dc_name": [
    null,
    null
  ],
  "number_of_vms": [
    1,
    "1"
  ],
  "src_vm_id": [
    1947,
    "bd-clone-template"
  ],
  "provision_type": [
    "native_clone",
    "Native Clone"
  ],
  "linked_clone": [
    null,
    null
  ],
  "vm_name": "aab_rest_vm2",
  "pxe_server_id": [
    null,
    null
  ],
  "schedule_type": [
    "immediately",
    "Immediately on Approval"
  ],
  "vm_auto_start": [
    true,
    1
  ]
}

```

```

],
"schedule_time": "2015-05-09T14:25:28-04:00",
"retirement": [
  0,
  "Indefinite"
],
"retirement_warn": [
  604800,
  "1 Week"
],
"stateless": [
  false,
  0
],
"vlan": [
  "rhevm",
  "rhevm"
],
"disk_format": [
  "default",
  "Default"
],
"number_of_sockets": [
  1,
  "1"
],
"cores_per_socket": [
  1,
  "1"
],
"vm_memory": [
  "1024",
  "1024"
],
"network_adapters": [
  1,
  "1"
],
"placement_host_name": [
  null,
  null
],
"placement_ds_name": [
  null,
  null
],
"src_vm_nics": [

```

```

],
"src_vm_lans": [

```

```

],
"customize_enabled": [
  "enabled"
],
"src_ems_id": [
  105,
  "rhevm230"
],
"auto_approve": false,
"ws_values": {
  "request_id": "1001"
},
"ws_ems_custom_attributes": {

```

```

    },
    "ws_miq_custom_attributes": {
    }
  },
  "userid": "jdoe",
  "source_id": "1947",
  "source_type": "VmOrTemplate"
},
{
  "href": "http://localhost:3000/api/provision_requests/21",
  "id": "21",
  "description": "Provision from [bd-clone-template] to [aab_rest_vm3]",
  "approval_state": "pending_approval",
  "type": "MiqProvisionRequest",
  "created_on": "2015-05-08T18:25:32Z",
  "updated_on": "2015-05-08T18:25:32Z",
  "requester_id": "1",
  "requester_name": "jdoe",
  "request_type": "template",
  "request_state": "pending",
  "message": "VM Provisioning - Request Created",
  "status": "Ok",
  "options": {
    "use_pre_dialog": false,
    "request_type": "template",
    "miq_request_dialog_name": "miq_provision_redhat_dialogs_template",
    "owner_first_name": "John",
    "owner_last_name": "Doe",
    "owner_email": "jdoe@sample.com",
    "vm_tags": [
      62,
      58
    ],
    "addr_mode": [
      "static",
      "Static"
    ],
    "placement_cluster_name": [
      null,
      null
    ],
    "cluster_filter": [
      null,
      null
    ],
    "placement_auto": [
      true,
      1
    ],
    "placement_dc_name": [
      null,
      null
    ],
    "number_of_vms": [
      1,
      "1"
    ],
    "src_vm_id": [
      1947,
      "bd-clone-template"
    ],
    "provision_type": [
      "native_clone",
      "Native Clone"
    ]
  }
}

```

```

],
"linked_clone": [
  null,
  null
],
"vm_name": "aab_rest_vm3",
"pxe_server_id": [
  null,
  null
],
"schedule_type": [
  "immediately",
  "Immediately on Approval"
],
"vm_auto_start": [
  true,
  1
],
"schedule_time": "2015-05-09T14:25:31-04:00",
"retirement": [
  0,
  "Indefinite"
],
"retirement_warn": [
  604800,
  "1 Week"
],
"stateless": [
  false,
  0
],
"vlan": [
  "rhevm",
  "rhevm"
],
"disk_format": [
  "default",
  "Default"
],
"number_of_sockets": [
  1,
  "1"
],
"cores_per_socket": [
  1,
  "1"
],
"vm_memory": [
  "1024",
  "1024"
],
"network_adapters": [
  1,
  "1"
],
"placement_host_name": [
  null,
  null
],
"placement_ds_name": [
  null,
  null
],
"src_vm_nics": [

```

```

    ],
    "src_vm_lans": [
    ],
    "customize_enabled": [
        "enabled"
    ],
    "src_ems_id": [
        105,
        "rhevm230"
    ],
    "auto_approve": false,
    "ws_values": {
        "request_id": "1001"
    },
    "ws_ems_custom_attributes": {
    },
    "ws_miq_custom_attributes": {
    }
    },
    "userid": "jdoe",
    "source_id": "1947",
    "source_type": "VmOrTemplate"
    }
    ]
}

```

Infrastructure Management API Examples - Service Catalogs

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Add a Service Catalog](#)
- [Add multiple Service Catalogs](#)
- [Edit a Service Catalog](#)
- [Edit Service Catalogs](#)
- [Assign Service Templates](#)
- [Unassign Service Templates](#)
- [Delete Service Catalogs](#)

- [Adding a simple Service Catalog](#)
 - [Request:](#)
 - [Response:](#)

Adding a simple Service Catalog

Request:

POST /api/service_catalogs

```

{
  "action" : "create",
  "resource" : {
    "name" : "Sample Service Catalog",

```

```

    "description" : "Description of Sample Service Catalog",
    "service_templates" : [
      { "href" : "http://localhost:3000/api/service_templates/3" },
      { "href" : "http://localhost:3000/api/service_templates/4" }
    ]
  }
}

```

Response:

```

{
  "results": [
    {
      "id": "7",
      "href" : "http://localhost:3000/api/service_catalogs/7",
      "name": "Sample Service Catalog",
      "description": "Description of Sample Service Catalog"
    }
  ]
}

```

- [Adding multiple Service Catalogs](#)
 - [Request:](#)
 - [Response:](#)

Adding multiple Service Catalogs

Request:

POST /api/service_catalogs

```

{
  "action" : "create",
  "resources" : [
    {
      "name" : "First Sample Service Catalog",
      "description" : "Description of First Sample Service Catalog",
      "service_templates" : [
        { "href" : "http://localhost:3000/api/service_templates/1" },
        { "href" : "http://localhost:3000/api/service_templates/2" }
      ]
    },
    {
      "name" : "Second Sample Service Catalog",
      "description" : "Description of Second Sample Service Catalog",
      "service_templates" : [
        { "href" : "http://localhost:3000/api/service_templates/3" },
        { "href" : "http://localhost:3000/api/service_templates/4" }
      ]
    },
    {
      "name" : "Third Sample Service Catalog",
      "description" : "Description of Third Sample Service Catalog",
      "service_templates" : [
        { "href" : "http://localhost:3000/api/service_templates/5" },
        { "href" : "http://localhost:3000/api/service_templates/6" }
      ]
    }
  ]
}

```


Response:

```
{
  "results": [
    {
      "id": "8",
      "href" : "http://localhost:3000/api/service_catalogs/8",
      "name": "First Sample Service Catalog",
      "description": "Description of First Sample Service Catalog"
    },
    {
      "id": "9",
      "href" : "http://localhost:3000/api/service_catalogs/9",
      "name": "Second Sample Service Catalog",
      "description": "Description of Second Sample Service Catalog"
    },
    {
      "id": "10",
      "href" : "http://localhost:3000/api/service_catalogs/10",
      "name": "Third Sample Service Catalog",
      "description": "Description of Third Sample Service Catalog"
    }
  ]
}
```

- [Edit Service Catalog](#)
 - [Request:](#)
 - [Response:](#)

Edit Service Catalog

Request:

POST /api/service_catalog/7

```
{
  "action" : "edit",
  "resource" : {
    "description" : "Updated Description of the Seventh Service Catalog"
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/service_catalogs/7",
  "id": "7",
  "name": "Sample Service Catalog",
  "description": "Updated Description of the Seventh Service Catalog",
  "service_templates": {
    "count": 1,
    "resources": [
      {
        "href": "http://localhost:3000/api/service_catalogs/7/service_templates/3"
      }
    ]
  }
}
```

- [Edit Multiple Service Catalogs](#)
 - [Request:](#)

- [Response:](#)

Edit Multiple Service Catalogs

Request:

POST /api/service_catalogs

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_catalogs/3",
      "description" : "Updated Description for Third Service Catalog"
    },
    {
      "href" : "http://localhost:3000/api/service_catalogs/6",
      "description" : "Updated Description for Sixth Service Catalog"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "id" : "3",
      "href" : "http://localhost:3000/api/service_catalogs/3",
      "name": "Service Catalog B Added from REST API",
      "description": "Updated Description for Third Service Catalog"
    },
    {
      "id" : "6",
      "href" : "http://localhost:3000/api/service_catalogs/6",
      "name": "Service Catalog E Added from REST API",
      "description": "Updated Description for Sixth Service Catalog"
    }
  ]
}
```

- [Assign service templates to Service Catalog 1](#)
 - [Request:](#)
 - [Response:](#)

Assign service templates to Service Catalog 1

Request:

POST /api/service_catalogs/1/service_templates

```
{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/1" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Service Template id:5 name:'template5'",
      "service_template_id": "5",
      "service_template_href": "http://localhost:3000/api/service_templates/5",
      "href": "http://localhost:3000/api/service_catalogs/1"
    },
    {
      "success": false,
      "message": "Service Template 1 is currently assigned to Service Catalog 2",
      "service_template_id": "1",
      "service_template_href": "http://localhost:3000/api/service_templates/1",
      "href": "http://localhost:3000/api/service_catalogs/1"
    }
  ]
}
```

- [Unassign service templates from Service Catalog 2](#)
 - [Request:](#)
 - [Response:](#)

Unassign service templates from Service Catalog 2

Request:

POST /api/service_catalogs/6/service_templates

```
{
  "action" : "unassign",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/1" },
    { "href" : "http://localhost:3000/api/service_templates/5" },
    { "href" : "http://localhost:3000/api/service_templates/8" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": false,
      "message": "Service Template 1 is not currently assigned to Service Catalog 3",
      "service_template_id": "1",
      "service_template_href": "http://localhost:3000/api/service_templates/1",
      "href": "http://localhost:3000/api/service_catalogs/6"
    },
    {
      "success": true,
      "message": "Unassigning Service Template id:5 name:'template5'",
      "service_template_id": "5",
      "service_template_href": "http://localhost:3000/api/service_templates/5",
      "href": "http://localhost:3000/api/service_catalogs/6"
    },
    {
      "success": false,
```

```

    "message": "Couldn't find ServiceTemplate with id=8",
    "href": "http://localhost:3000/api/service_catalogs/6"
  }
]
}

```

- [Delete multiple service catalogs](#)
 - [Request:](#)
 - [Response:](#)

Delete multiple service catalogs

Request:

POST /api/service_catalogs

```

{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_catalogs/8" },
    { "href" : "http://localhost:3000/api/service_catalogs/9" },
    { "href" : "http://localhost:3000/api/service_catalogs/10" }
  ]
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "service_catalogs id: 8 deleting",
      "href": "http://localhost:3000/api/service_catalogs/8"
    },
    {
      "success": true,
      "message": "service_catalogs id: 9 deleting",
      "href": "http://localhost:3000/api/service_catalogs/9"
    },
    {
      "success": true,
      "message": "service_catalogs id: 10 deleting",
      "href": "http://localhost:3000/api/service_catalogs/10"
    }
  ]
}

```

Infrastructure Management API Examples - Service Templates

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Edit a Service Template](#)
- [Edit Service Templates](#)
- [Delete Service Templates](#)

- [Edit Service Template](#)
 - [Request:](#)
 - [Response:](#)

Edit Service Template

Request:

POST /api/service_templates/2

```
{
  "action" : "edit",
  "resource" : {
    "name" : "updated_svc_template_02",
    "description" : "This is an updated description for service template 02"
  }
}
```

Response:

```
{
  "href": "http://localhost:3000/api/service_templates/2",
  "id": "2",
  "name": "updated_svc_template_02",
  "description": "This is an updated description for service template 02",
  "guid": "6f7918b4-d6e7-11e4-9837-b8e85646e742",
  "options": {
  },
  "created_at": "2015-03-30T14:17:02Z",
  "updated_at": "2015-04-16T22:30:02Z",
  "service_type": "unknown",
  "service_template_catalog_id": "6"
}
```

- [Edit Service Templates](#)
 - [Request:](#)
 - [Response:](#)

Edit Service Templates

Request:

POST /api/service_templates

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_templates/1",
      "description" : "This is an updated description for the first sample service template"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/2",
      "description" : "This is an updated description for the second sample service template"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "id": "1",
      "name": "template1",
      "description": "This is an updated description for the first sample service
template",
      "guid": "6a6fdf7e-d6e7-11e4-9837-b8e85646e742",
      "options": {
      },
      "created_at": "2015-03-30T14:16:53Z",
      "updated_at": "2015-04-16T22:33:09Z",
      "service_type": "unknown",
      "service_template_catalog_id": "3"
    },
    {
      "id": "2",
      "name": "updated_svc_template_02",
      "description": "This is an updated description for the second sample service
template",
      "guid": "6f7918b4-d6e7-11e4-9837-b8e85646e742",
      "options": {
      },
      "created_at": "2015-03-30T14:17:02Z",
      "updated_at": "2015-04-16T22:33:09Z",
      "service_type": "unknown",
      "service_template_catalog_id": "6"
    }
  ]
}
```

- [Delete multiple service templates](#)
 - [Request:](#)
 - [Response:](#)

Delete multiple service templates

Request:

POST /api/service_templates

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/service_templates/4" },
    { "href" : "http://localhost:3000/api/service_templates/5" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "service_templates id: 4 deleting",
      "href": "http://localhost:3000/api/service_templates/4"
    },
    {

```

```

    "success": true,
    "message": "service_templates id: 5 deleting",
    "href": "http://localhost:3000/api/service_templates/5"
  }
}
1
}

```

Infrastructure Management API Examples - Services

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Order a Service](#)
- [Order Multiple Services](#)
- [Edit a Service](#)
- [Edit a Service with Patch Method](#)
- [Edit a Service with PUT](#)
- [Edit Services](#)
- [Reconfigure a Service](#)
- [Refresh Service Template Dialog Fields](#)
- [Refresh Service Dialog Fields](#)
- [Retire a Service](#)
- [Retire Multiple Services](#)
- [Trigger a Custom Action on a Service](#)
- [Delete Services](#)
- [Order a single service from Service Catalog 1](#)
 - [Request:](#)
 - [Response:](#)

Order a single service from Service Catalog 1

Request:

POST /api/service_catalogs/1/service_templates

```

{
  "action" : "order",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/3",
    "option_0_vm_target_name" : "test-vm-0001",
    "option_0_vm_target_hostname" : "test-vm-0001"
  }
}

```

Response:

```

{
  results: [
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",
      "description": "Provisioning Service [aws-ubuntu-api] from [aws-ubuntu-api]",
      "destination_id": null,
      "destination_type": null,

```

```

    "fulfilled_on": null,
    "id": "13",
    "message": "Service_Template_Provisioning - Request Created",
    "options": {
      "dialog": {
        "dialog_option_0_vm_target_name" : "test-vm-0001",
        "dialog_option_0_vm_target_hostname" : "test-vm-0001"
      }
    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
  }
]
}

```

- [Order multiple services from Service Catalog 2](#)
 - [Request:](#)
 - [Response:](#)

Order multiple services from Service Catalog 2

Request:

POST /api/service_catalogs/2/service_templates

```

{
  "action" : "order",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/service_templates/3",
      "option_1_vm_target_name" : "sample-vm-1201",
      "option_2_vm_target_hostname" : "sample-vm-1201"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/3",
      "option_1_vm_target_name" : "sample-vm-1202",
      "option_2_vm_target_hostname" : "sample-vm-1202"
    },
    {
      "href" : "http://localhost:3000/api/service_templates/4",
      "option_1_vm_target_name" : "dev-vm1",
      "option_2_vm_target_hostname" : "dev-vm1",
      "option_3_vm_memory" : '16384'
    }
  ],
}

```

Response:

```

{
  results: [
    {
      "approval_state": "pending_approval",
      "created_on": "2014-07-02T19:28:12Z",

```



```

    "description": "Provisioning Service [sample-vm-1201] from [sample-vm-
1201]",
    "destination_id": null,
    "destination_type": null,
    "fulfilled_on": null,
    "id": "13",
    "message": "Service_Template_Provisioning - Request Created",
    "options": {
      "dialog": {
        "dialog_option_0_vm_target_name" : "test-vm-0001",
        "dialog_option_0_vm_target_hostname" : "test-vm-0001"
      }
    }
  },
  "request_state": "pending",
  "request_type": "clone_to_service",
  "requester_id": "1",
  "requester_name": "Administrator",
  "source_id": "6",
  "source_type": "ServiceTemplate",
  "status": "Ok",
  "updated_on": "2014-07-02T19:28:12Z",
  "userid": "admin"
},
{
  "approval_state": "pending_approval",
  "created_on": "2014-07-02T19:28:12Z",
  "description": "Provisioning Service [sample-vm-1202] from [sample-vm-
1202]",
  "destination_id": null,
  "destination_type": null,
  "fulfilled_on": null,
  "id": "13",
  "message": "Service_Template_Provisioning - Request Created",
  "options": {
    "dialog": {
      "dialog_option_0_vm_target_name" : "test-vm-0001",
      "dialog_option_0_vm_target_hostname" : "test-vm-0001"
    }
  }
},
  "request_state": "pending",
  "request_type": "clone_to_service",
  "requester_id": "1",
  "requester_name": "Administrator",
  "source_id": "6",
  "source_type": "ServiceTemplate",
  "status": "Ok",
  "updated_on": "2014-07-02T19:28:12Z",
  "userid": "admin"
},
{
  "approval_state": "pending_approval",
  "created_on": "2014-07-02T19:28:12Z",
  "description": "Provisioning Service [sample-vm-1201] from [sample-vm-
1201]",
  "destination_id": null,
  "destination_type": null,
  "fulfilled_on": null,
  "id": "13",
  "message": "Service_Template_Provisioning - Request Created",
  "options": {
    "dialog": {
      "dialog_option_0_vm_target_name" : "test-vm-0001",
      "dialog_option_0_vm_target_hostname" : "test-vm-0001"
    }
  }
}

```

```

    }
    "request_state": "pending",
    "request_type": "clone_to_service",
    "requester_id": "1",
    "requester_name": "Administrator",
    "source_id": "6",
    "source_type": "ServiceTemplate",
    "status": "Ok",
    "updated_on": "2014-07-02T19:28:12Z",
    "userid": "admin"
  }
]
}

```

- [Edit Service](#)
 - [Request:](#)
 - [Response:](#)

Edit Service

Request:

POST /api/services/101

```

{
  "action" : "edit",
  "resource" : {
    "name" : "service_101",
    "description" : "This is an updated description for service 101"
  }
}

```

Response:

```

{
  "href": "http://localhost:3000/api/services/101",
  "id": "101",
  "name": "service_101",
  "description": "This is an updated description for the service 101",
  "guid": "4a8f96de-ala6-11e4-9f8d-b8e85646e742",
  "options": {
  },
  "created_at": "2015-01-21T19:47:11Z",
  "updated_at": "2015-04-16T22:17:15Z"
}

```

- [Editing a resource via the Patch Method.](#)
 - [Request:](#)
 - [Response:](#)

Editing a resource via the Patch Method.

Supported attribute actions for PATCH include, add, edit and remove.

Request:

PATCH /api/services/90

```

[
  { "action" : "edit", "path" : "name", "value" : "updated_service_90" },

```

```
{ "action" : "remove", "path" : "description"}
]
```

Response:

```
{
  "href": "http://localhost:3000/api/services/90",
  "id": "90",
  "name": "updated_service_90",
  "guid": "eb6fc61c-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-04-16T22:38:57Z"
}
```

Note:

Note that the description attribute is no longer defined for this service.

- [Edit Service via PUT](#)
 - [Request:](#)
 - [Response:](#)

Edit Service via PUT

Request:

PUT /api/services/90

```
{
  "name" : "new_service_90",
  "description" : "This is a new description for the service 90"
}
```

Response:

```
{
  "href": "http://localhost:3000/api/services/90",
  "id": "90",
  "name": "new_service_90",
  "description": "This is a new description for the service 90",
  "guid": "eb6fc61c-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-04-16T22:41:37Z"
}
```

- [Edit Services](#)
 - [Request:](#)
 - [Response:](#)

Edit Services

Request:

POST /api/services

```
{
  "action" : "edit",
  "resources" : [
    {
      "href" : "http://localhost:3000/api/services/81",
      "description" : "This is an updated description for service 81"
    },
    {
      "href" : "http://localhost:3000/api/services/82",
      "description" : "This is an updated description for service 82"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/services/81",
      "id": "81",
      "name": "api_gen_A_81",
      "description": "This is an updated description for service 81",
      "guid": "eb6daaf8-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-04-16T22:43:37Z"
    },
    {
      "href": "http://localhost:3000/api/services/82",
      "id": "82",
      "name": "api_gen_A_82",
      "description": "This is an updated description for service 82",
      "guid": "eb6de400-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-04-16T22:43:37Z"
    }
  ]
}
```

- [Reconfigure Service](#)
 - [Request:](#)
 - [Response:](#)

Reconfigure Service

Request:

POST /api/services/2

```
{
  "action" : "reconfigure",
  "resource" : {
    "SampleText" : "default_12",
    "SampleHostIp" : "100.200.300.12"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Service id:2 name:'SampleStaticServiceCatalogItem' reconfiguring",
  "result": {
    "id": "74",
    "description": "Service Reconfigure for: SampleStaticServiceCatalogItem",
    "approval_state": "pending_approval",
    "type": "ServiceReconfigureRequest",
    "created_on": "2016-06-10T20:34:13Z",
    "updated_on": "2016-06-10T20:34:13Z",
    "requester_id": "1",
    "requester_name": "Administrator",
    "request_type": "service_reconfigure",
    "request_state": "pending",
    "message": "Service Reconfigure - Request Created",
    "status": "Ok",
    "options": {
      "dialog": {
        "dialog_SampleText": "default_12",
        "dialog_SampleHostName": "",
        "dialog_SampleHostIp": "100.200.300.12",
        "dialog_sample_button_one": "complex_stuff",
        "dialog_vm_host_name": ""
      },
      "workflow_settings": {
        "resource_action_id": "23",
        "dialog_id": "3"
      },
      "src_id": "2",
      "requester_group": "EvmGroup-super_administrator"
    },
    "userid": "admin",
    "source_id": "2",
    "source_type": "Service",
    "tenant_id": "1",
    "service_order_id": "9",
    "process": true
  },
  "href": "http://localhost:3000/api/services/2"
}
```

- [Refresh Dialog Fields](#)
 - [Request:](#)
 - [Response:](#)

Refresh Dialog Fields

Refresh fields of a service template dialog:

Request:

POST /api/service_catalogs/1/service_templates

```
{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "href" : "http://localhost:3000/api/service_templates/1",
    "dialog_fields" : {
      "my_field" : "current value for my_field"
    }
  }
}
```

```

    "fields" : [ "my_field" ]
  }
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "Refreshing dialog fields for Service Template id:1
name: 'DynamicDialog'",
      "result": {
        "my_field": {
          "name": "my_field",
          "description": "my_field",
          "type": "DialogFieldDropDownList",
          "data_type": "string",
          "display": "edit",
          "display_method_options": {
          },
          "required": true,
          "required_method_options": {
          },
          "values": [
            [
              "",
              "Choose"
            ],
            [
              "6",
              "6"
            ],
            [
              "15",
              "15"
            ],
            [
              "2016-06-10T15:14:30Z",
              "2016-06-10T15:14:30Z"
            ]
          ],
          "values_method_options": {
          },
          "options": {
            "sort_by": "value",
            "sort_order": "ascending"
          },
          "label": "My Field",
          "position": 0,
          "dynamic": true,
          "show_refresh_button": true,
          "load_values_on_init": true,
          "read_only": false,
          "resource_action": {
            "resource_type": "DialogField",
            "ae_namespace": "Dialogs",
            "ae_class": "dynamic_dialogs",
            "ae_instance": "drop_down",
            "ae_attributes": {
            }
          }
        }
      }
    }
  ]
}

```

```

    },
    "service_template_id": "1",
    "service_template_href": "http://localhost:3000/api/service_templates/1",
    "href": "http://localhost:3000/api/service_catalogs/1"
  }
]
}

```

- [Refresh Dialog Fields](#)
 - [Request:](#)
 - [Response:](#)

Refresh Dialog Fields

Refresh fields of a service dialog:

Request:

POST /api/service_dialogs/2

```

{
  "action" : "refresh_dialog_fields",
  "resource" : {
    "dialog_fields" : {
      "my_field" : "current value for my_field"
    },
    "fields" : [ "my_field" ]
  }
}

```

Response:

```

{
  "success": true,
  "message": "Refreshing dialog fields for Service Dialog id:2 label:'Dynamic Drop-down test'",
  "result": {
    "my_field": {
      "name": "my_field",
      "description": "my_field",
      "type": "DialogFieldDropDownList",
      "data_type": "string",
      "display": "edit",
      "display_method_options": {
      },
      "required": true,
      "required_method_options": {
      },
      "values": [
        [
          "",
          "Choose"
        ],
        [
          "7",
          "7"
        ],
        [
          "13",
          "13"
        ]
      ],
    }
  }
}

```

```

    [
      "2016-06-10T17:59:36Z",
      "2016-06-10T17:59:36Z"
    ]
  ],
  "values_method_options": {
  },
  "options": {
    "sort_by": "value",
    "sort_order": "ascending"
  },
  "label": "My Field",
  "position": 0,
  "dynamic": true,
  "show_refresh_button": true,
  "load_values_on_init": true,
  "read_only": false,
  "resource_action": {
    "resource_type": "DialogField",
    "ae_namespace": "Dialogs",
    "ae_class": "dynamic_dialogs",
    "ae_instance": "drop_down",
    "ae_attributes": {
    }
  }
}
},
"href": "http://localhost:3000/api/service_dialogs/2"
}

```

- [Retire Service Immediately](#)
 - [Request:](#)
 - [Response:](#)
 - [Response:](#)
- [Retire Service In the Future](#)
 - [Request:](#)
 - [Response:](#)

Retire Service Immediately

Request:

POST /api/services

```

{
  "action" : "request_retire",
  "resource" : { "href" : "http://localhost:3000/api/services/35" }
}

```

Response:

```

{
  "results" : [
    {
      "href": "http://localhost:3000/api/services/95",
      "id": "95",
      "name": "sample_service_95",
      "description": "Description for sample_service_95",
      "guid": "eb713538-7c9b-11e4-8a3a-b8e85646e742",
      "options": {

```



```

    },
    "created_at": "2014-12-05T16:29:43Z",
    "updated_at": "2015-05-08T19:32:00Z"
  }
]
}

```

Or simply targeting the service directly:

POST /api/services/95

```

{
  "action" : "request_retire"
}

```

Response:

```

{
  "href": "http://localhost:3000/api/services/95",
  "id": "95",
  "name": "sample_service_95",
  "description": "Description for sample_service_95",
  "guid": "eb713538-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-05-08T19:32:00Z"
}

```

Retire Service In the Future

Request:

POST /api/services/93

```

{
  "action" : "request_retire",
  "resource" : { "date" : "10/31/2015", "warn" : "7" }
}

```

Response:

```

{
  "href": "http://localhost:3000/api/services/93",
  "id": "93",
  "name": "sample_service_93",
  "description": "Description for sample_service_93",
  "guid": "eb709e02-7c9b-11e4-8a3a-b8e85646e742",
  "options": {
  },
  "created_at": "2014-12-05T16:29:43Z",
  "updated_at": "2015-05-08T19:40:19Z",
  "retired": false,
  "retires_on": "2015-10-30",
  "retirement_warn": 7
}

```

- [Retire Multiple Services](#)
 - [Request:](#)
 - [Response:](#)

Retire Multiple Services

Request:

POST /api/services

```
{
  "action" : "request_retire",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/100" },
    { "href" : "http://localhost:3000/api/services/101", "date" : "11/01/2015",
"warn" : "4" },
    { "href" : "http://localhost:3000/api/services/102", "date" : "11/02/2015",
"warn" : "4" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/services/100",
      "id": "100",
      "name": "cloud_service_100",
      "description": "Description for cloud_service_100",
      "guid": "eb725f94-7c9b-11e4-8a3a-b8e85646e742",
      "options": {
      },
      "created_at": "2014-12-05T16:29:43Z",
      "updated_at": "2015-05-08T19:46:11Z"
    },
    {
      "href": "http://localhost:3000/api/services/101",
      "id": "101",
      "name": "cloud_service_101",
      "description": "Description for cloud_service_101",
      "guid": "4a8f96de-a1a6-11e4-9f8d-b8e85646e742",
      "options": {
      },
      "created_at": "2015-01-21T19:47:11Z",
      "updated_at": "2015-05-08T19:46:19Z",
      "retired": false,
      "retires_on": "2015-11-01",
      "retirement_warn": 4
    },
    {
      "href": "http://localhost:3000/api/services/102",
      "id": "102",
      "name": "cloud_service_102",
      "description": "Description for cloud_service_102",
      "guid": "e2da0cb0-e47e-11e4-a47f-b8e85646e742",
      "options": {
      },
      "created_at": "2015-04-16T21:23:55Z",
      "updated_at": "2015-05-08T19:46:19Z",
      "retired": false,
      "retires_on": "2015-11-02",
      "retirement_warn": 3
    }
  ]
}
```

- [Trigger a Custom Action on a Service](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)

Trigger a Custom Action on a Service

This page shows an ordered service that has a custom button with a dialog and how the custom action is triggered.

Here is the ordered service with the expanded custom actions:

Request:

```
GET /api/services/110?attributes=custom_actions
```

Response:

```
{
  "href": "http://localhost:3000/api/services/110",
  "id": "110",
  "name": "SampleServiceTemplate",
  "description": "Sample Service Template",
  "guid": "a3ab175c-8ca1-11e5-939f-000a27020067",
  "service_template_id": "9",
  "options": {
    "button_order": [
      "cb-10"
    ],
    "dialog": {
      "dialog_hostname": "myhostname",
      "dialog_ipaddress": "192.168.100.1",
      "dialog_environmenttype": "medium"
    }
  },
  "display": true,
  "created_at": "2015-11-16T20:35:56Z",
  "updated_at": "2015-11-16T20:35:56Z",
  "evm_owner_id": "1",
  "miq_group_id": "1",
  "tenant_id": "1",
  "custom_actions": {
    "buttons": [
      {
        "id": "10",
        "guid": "cdcec41c-8ca0-11e5-8c79-000a27020067",
        "description": "Query VM",
        "applies_to_class": "ServiceTemplate",
        "options": {
          "button_image": 4,
          "display": true
        },
        "userid": "admin",
        "created_on": "2015-11-16T20:29:58Z",
        "updated_on": "2015-11-16T20:29:58Z",
        "name": "QueryVM",
        "visibility": {
          "roles": [
            "_ALL_"
          ]
        }
      }
    ]
  }
}
```

```

    ],
    "applies_to_id": "9",
    "resource_action": {
        "id": "38",
        "dialog_id": "4",
        "resource_id": "10",
        "resource_type": "CustomButton",
        "created_at": "2015-11-16T20:29:58Z",
        "updated_at": "2015-11-16T20:29:58Z",
        "ae_namespace": "SYSTEM",
        "ae_class": "PROCESS",
        "ae_instance": "Request",
        "ae_attributes": {
            "attr1": "value1",
            "attr2": "value2",
            "request": "System/Project/InspectME"
        }
    },
    "dialog": {
        "description": "Sample Dialog Description",
        "buttons": "submit",
        "label": "SampleDialog",
        "dialog_tabs": [
            {
                "description": "VM Tab Description",
                "display": "edit",
                "label": "VMTabLabel",
                "position": 0,
                "dialog_groups": [
                    {
                        "description": "Vm Info Description",
                        "display": "edit",
                        "label": "VmInfoLabel",
                        "position": 0,
                        "dialog_fields": [
                            {
                                "name": "hostname",
                                "description": "Hostname of the VM",
                                "type": "DialogFieldTextBox",
                                "display": "edit",
                                "display_method_options": {
                                },
                                "required": false,
                                "required_method_options": {
                                },
                                "default_value": "default_hostname",
                                "values_method_options": {
                                },
                                "options": {
                                    "protected": false
                                },
                                "label": "HostNameLabel",
                                "position": 0,
                                "dynamic": false,
                                "read_only": false,
                                "resource_action": {
                                    "resource_type": "DialogField",
                                    "ae_attributes": {
                                    }
                                }
                            }
                        ]
                    }
                ],
                "name": "ipaddress",
                "description": "IP Address of VM",
            }
        ]
    }
}

```

```

        "type": "DialogFieldTextBox",
        "display": "edit",
        "display_method_options": {
        },
        "required": false,
        "required_method_options": {
        },
        "default_value": "",
        "values_method_options": {
        },
        "options": {
            "protected": false
        },
        "label": "IpaddressLabel",
        "position": 1,
        "dynamic": false,
        "read_only": false,
        "resource_action": {
            "resource_type": "DialogField",
            "ae_attributes": {
            }
        }
    }
}
]
}
},
{
    "description": "Environment Tab Description",
    "display": "edit",
    "label": "EnvironmentTab",
    "position": 1,
    "dialog_groups": [
        {
            "description": "Environment Box Description",
            "display": "edit",
            "label": "EnvironmentBoxLabel",
            "position": 0,
            "dialog_fields": [
                {
                    "name": "environmenttype",
                    "description": "Environment Type Description",
                    "type": "DialogFieldRadioButton",
                    "data_type": "string",
                    "display": "edit",
                    "display_method_options": {
                    },
                    "required": true,
                    "required_method_options": {
                    },
                    "values": [
                        [
                            "large",
                            "Large Environment"
                        ],
                        [
                            "medium",
                            "Medium Environment"
                        ],
                        [
                            "small",
                            "Small Environment"
                        ]
                    ]
                }
            ]
        }
    ],

```

```

        "values_method_options": {
        },
        "options": {
            "sort_by": "description",
            "sort_order": "ascending"
        },
        "label": "EnvTypeLabel",
        "position": 0,
        "dynamic": false,
        "read_only": false,
        "resource_action": {
            "resource_type": "DialogField",
            "ae_attributes": {
            }
        }
    }
}
]
}
]
}
]
}
}
]
},
"button_groups": [
]
},
"actions": [
    {
        "name": "edit",
        "method": "post",
        "href": "http://localhost:3000/api/services/110"
    },
    {
        "name": "request_retire",
        "method": "post",
        "href": "http://localhost:3000/api/services/110"
    },
    {
        "name": "set_ownership",
        "method": "post",
        "href": "http://localhost:3000/api/services/110"
    },
    {
        "name": "delete",
        "method": "delete",
        "href": "http://localhost:3000/api/services/110"
    },
    {
        "name": "queryvm",
        "method": "post",
        "href": "http://localhost:3000/api/services/110"
    }
]
}

```

Showing the **queryvm** custom action being returned in the actions list.

The following is the request triggering a **queryvm** action on the service:

Request:

POST /api/services/110

```
{
  "action" : "queryvm",
  "resource" : {
    "hostname": "updated_hostname",
    "ipaddress": "192.168.100.2",
    "environmenttype" : "medium"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Invoked custom dialog action queryvm for services id: 110",
  "href": "http://localhost:3000/api/services/110"
}
```

- [Delete Services](#)
 - [Request:](#)
 - [Response:](#)

Delete Services

Request:

POST /api/services

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/97" },
    { "href" : "http://localhost:3000/api/services/98" },
    { "href" : "http://localhost:3000/api/services/99" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "services id: 97 deleting",
      "href": "http://localhost:3000/api/services/97"
    },
    {
      "success": true,
      "message": "services id: 98 deleting",
      "href": "http://localhost:3000/api/services/98"
    },
    {
      "success": true,
      "message": "services id: 99 deleting",
      "href": "http://localhost:3000/api/services/99"
    }
  ]
}
```

Infrastructure Management API Examples - VM Management

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Start a VM](#)
- [Stop a VM](#)
- [Suspend a VM](#)
- [Scan a VM](#)
- [Set owner of a VM](#)
- [Add Event to a VM](#)
- [Add Lifecycle Event to VM](#)
- [Delete a VM](#)
- [Start a VM](#)
 - [Request:](#)
 - [Response:](#)
- [Querying task progress:](#)
 - [Request:](#)
 - [Response:](#)

Start a VM

Request:

```
POST /api/vms/1921

{
  "action": "start"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' starting",
  "task_id": "610",
  "task_href": "http://localhost:3000/api/tasks/610",
  "href": "http://localhost:3000/api/vms/1921"
}
```

Querying task progress:

Request:

```
GET /api/tasks/620
```

Response:

```
{
  "href": "http://localhost:3000/api/tasks/610",
  "id": "610",
  "name": "VM id:1921 name:'aab_demo_vm' starting",
}
```



```

    "state": "Queued",
    "status": "Ok",
    "message": "Queued the action: [VM id:1921 name:'aab_demo_vm' starting] being
run for user: [admin]",
    "userid": "admin",
    "created_on": "2015-05-05T15:58:08Z",
    "updated_on": "2015-05-05T15:58:08Z"
}

```

- [Stop a VM](#)
 - [Request:](#)
 - [Response:](#)

Stop a VM

Request:

POST /api/vms/1921

```

{
  "action": "stop"
}

```

Response:

```

{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' stopping",
  "task_id": "619",
  "task_href": "http://localhost:3000/api/tasks/619",
  "href": "http://localhost:3000/api/vms/1921"
}

```

- [Suspend a VM](#)
 - [Request:](#)
 - [Response:](#)

Suspend a VM

Request:

POST /api/vms/1921

```

{
  "action": "suspend"
}

```

Response:

```

{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' suspending",
  "task_id": "620",
  "task_href": "http://localhost:3000/api/tasks/620",
  "href": "http://localhost:3000/api/vms/1921"
}

```

- [Scan a VM](#)
 - [Request:](#)

- [Response:](#)
- [Response:](#)

Scan a VM

Request:

POST /api/vms/1922

```
{
  "action": "scan"
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1922 name:'aab_test_vm' scanning",
  "task_id": "618",
  "task_href": "http://localhost:3000/api/tasks/618",
  "href": "http://localhost:3000/api/vms/1922"
}
```

And then optionally waiting for the task to complete as follows:

GET /api/tasks/618

Response:

```
{
  "href": "http://localhost:3000/api/tasks/618",
  "id": "618",
  "name": "VM id:1922 name:'aab_test_vm' scanning",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-05T19:37:32Z",
  "updated_on": "2015-05-05T19:37:38Z"
}
```

- [Set owner of a VM](#)
 - [Request:](#)
 - [Response:](#)

Set owner of a VM

Request:

POST /api/vms/1921

```
{
  "action": "set_owner",
  "resource" : {
    "owner" : "admin"
  }
}
```

Response:

```
{
  "success": true,
  "message": "VM id:1921 name:'aab_demo_vm' setting owner to 'admin'",
  "href": "http://localhost:3000/api/vms/1921"
}
```

- [Add event to a VM](#)
 - [Request:](#)
 - [Response:](#)

Add event to a VM

Request:

POST /api/vms/1921

```
{
  "action": "add_event",
  "resource" : {
    "event_type" : "BadUserNameSessionEvent",
    "event_message" : "Cannot login user@test.domain"
  }
}
```

Response:

```
{
  "success": true,
  "message": "Adding Event type=BadUserNameSessionEvent message=Cannot login user@test.domain",
  "href": "http://localhost:3000/api/vms/1921"
}
```

- [Delete VMs](#)
 - [Deleting a single VM](#)
 - [Request:](#)
 - [Deleting multiple VMs](#)
 - [Request:](#)
 - [Response:](#)
 - [Request:](#)
 - [Response:](#)

Delete VMs

Deleting a single VM

Request:

DELETE /api/vms/334

Upon success, Vm targeted for deletion asynchronously and No Response Content with HTTP status code 204

Deleting multiple VMs

Request:

DELETE /api/vms

```
{
  "action" : "delete",
  "resources" : [
    { "href" : "http://localhost:3000/api/vms/348" },
    { "href" : "http://localhost:3000/api/vms/349" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "VM id:348 name:'aab-temp1' deleting",
      "task_id": "616",
      "task_href": "http://localhost:3000/api/tasks/616",
      "href": "http://localhost:3000/api/vms/348"
    },
    {
      "success": true,
      "message": "VM id:349 name:'aab-temp2' deleting",
      "task_id": "617",
      "task_href": "http://localhost:3000/api/tasks/617",
      "href": "http://localhost:3000/api/vms/349"
    }
  ]
}
```

Optionally monitoring the asynchronous Vm deletion by accessing the related Task as follows:

Request:

GET /api/tasks/616

Response:

```
{
  "href": "http://localhost:3000/api/tasks/616",
  "id": "616",
  "name": "VM id:348 name:'aab-temp1' deleting",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2015-05-05T19:33:35Z",
  "updated_on": "2015-05-05T19:33:40Z"
}
```

Infrastructure Management API Examples - Tag Management

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Assign Tags](#)

- [Assign Tags to Service Template](#)
- [Unassign Tags](#)
- [Unassign Tags from Service Template](#)
- [Assign Tags to a Service](#)
 - [Request:](#)
 - [Response:](#)
- [Assign Tags by Name to a Service](#)
 - [Request:](#)
 - [Response:](#)
- [Assign a Tag by Reference to a Service](#)
 - [Request:](#)
 - [Response:](#)

Assign Tags to a Service

Request:

POST /api/services/101/tags

```
{
  "action" : "assign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" },
    { "category" : "environment", "name" : "dev" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'environment' name:'dev'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "environment",
      "tag_name": "dev"
    }
  ]
}
```

Assign Tags by Name to a Service

Request:

POST /api/services/101/tags

```
{
  "action" : "assign",
  "resources" : [
    { "name" : "/department/finance" },
    { "name" : "/location/ny" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "location",
      "tag_name": "ny"
    }
  ]
}
```

Assign a Tag by Reference to a Service

Request:

POST /api/services/101/tags

```
{
  "action" : "assign",
  "resources" : [
    { "href" : "http://localhost:3000/api/services/1/tags/49" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/49"
    }
  ]
}
```

- [Assign Tags to service template 1](#)
 - [Request:](#)
 - [Response:](#)

Assign Tags to service template 1

Request:

POST /api/service_templates/1/tags

```
{
  "action" : "assign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Assigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}
```

- [Unassign Tags from a Service](#)
 - [Request:](#)
 - [Response:](#)
- [Unassign a Tag by Name from a Service](#)
 - [Request:](#)
 - [Response:](#)
- [Unassign a Tag by Reference from a Service](#)
 - [Request:](#)
 - [Response:](#)

Unassign Tags from a Service

Request:

POST /api/services/101/tags

```
{
  "action" : "unassign",
  "resources" : [
```

```

    { "category" : "department", "name" : "finance" },
    { "category" : "environment", "name" : "dev" }
  ]
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'environment' name:'dev'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "environment",
      "tag_name": "dev"
    }
  ]
}

```

Unassign a Tag by Name from a Service

Request:

POST /api/services/101/tags

```

{
  "action" : "unassign",
  "resources" : [
    { "name" : "/managed/department/finance" }
  ]
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}

```

Unassign a Tag by Reference from a Service

Request:

POST /api/services/101/tags


```
{
  "action" : "unassign",
  "resources" : [
    { "href" : "tags/49" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/services/101",
      "tag_category": "department",
      "tag_name": "finance",
      "tag_href": "http://localhost:3000/api/tags/49"
    }
  ]
}
```

- [Unassign Tags from service template 1](#)
 - [Request:](#)
 - [Response:](#)

Unassign Tags from service template 1

Request:

POST /api/service_templates/1/tags

```
{
  "action" : "unassign",
  "resources" : [
    { "category" : "location", "name" : "ny" },
    { "category" : "department", "name" : "finance" }
  ]
}
```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Unassigning Tag: category:'location' name:'ny'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "location",
      "tag_name": "ny"
    },
    {
      "success": true,
      "message": "Unassigning Tag: category:'department' name:'finance'",
      "href": "http://localhost:3000/api/service_templates/1",
      "tag_category": "department",
      "tag_name": "finance"
    }
  ]
}
```

Infrastructure Management API Examples - Usage Flows

In this guide you will find examples for the Infrastructure Automation – Infrastructure Management REST API.

- [Compliance Policy Flow](#)
- [Compliance Policy Flow](#)
 - [Create a couple of conditions for the new compliance policy](#)
 - [Request:](#)
 - [Response:](#)
 - [Find the Policy action to use](#)
 - [Request:](#)
 - [Response:](#)
 - [Find the Policy event to use](#)
 - [Response:](#)
 - [Create the Compliance Policy for Finance Servers](#)
 - [Request:](#)
 - [Response:](#)
 - [Create the Compliance Policy Profile for Finance Servers](#)
 - [Request:](#)
 - [Response:](#)
 - [Assigning new policy to policy profile](#)
 - [Request:](#)
 - [Response:](#)
 - [Assigning the policy profile to a virtual machine](#)
 - [Request:](#)
 - [Response:](#)
 - [Querying the policy profiles and related policies of the virtual machine](#)
 - [Request:](#)
 - [Response:](#)
 - [Requesting a compliance check on the virtual machine](#)
 - [Request:](#)
 - [Response:](#)
 - [Waiting for the compliance check to complete](#)
 - [Response:](#)
 - [Querying all compliances run for the virtual machine](#)
 - [Request:](#)
 - [Response:](#)
 - [Querying the virtual machine and its latest compliance result including the last compliance status](#)
 - [Request:](#)
 - [Response:](#)

Compliance Policy Flow

Create a couple of conditions for the new compliance policy

Request:

POST /api/conditions

```
{
  "action" : "create",
  "resources" : [
    {
      "name" : "finance_mem_req",
      "description" : "System Memory Requirement for Finance Systems",
      "expression" : {
        ">=": {
          "field": "Vm-mem_cpu",
          "value": "4096"
        }
      },
      "towhat" : "Vm"
    },
    {
      "name" : "finance_cpu_req",
      "description" : "System CPU Requirement for Finance Systems",
      "expression" : {
        ">=": {
          "field": "Vm-num_cpu",
          "value": "2"
        }
      },
      "towhat" : "Vm"
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/conditions/8",
      "id": "8",
      "name": "finance_mem_req",
      "description": "System Memory Requirement for Finance Systems",
      "expression": {
        "exp": {
          ">=": {
            "field": "Vm-mem_cpu",
            "value": "4096"
          }
        }
      },
      "context_type": null,
      "towhat": "Vm",
      "file_mtime": null,
      "created_on": "2020-03-24T17:19:21Z",
      "updated_on": "2020-03-24T17:19:21Z",
      "guid": "b67a564d-5290-4245-b4bc-4d75d0890951",
      "filename": null,
      "applies_to_exp": null,
      "miq_policy_id": null,
      "notes": null,
      "read_only": null
    },
    {
      "href": "http://localhost:3000/api/conditions/9",

```

```

    "id": "9",
    "name": "finance_cpu_req",
    "description": "System CPU Requirement for Finance Systems",
    "expression": {
      "exp": {
        ">=": {
          "field": "Vm-num_cpu",
          "value": "2"
        }
      },
      "context_type": null
    },
    "towhat": "Vm",
    "file_mtime": null,
    "created_on": "2020-03-24T17:19:21Z",
    "updated_on": "2020-03-24T17:19:21Z",
    "guid": "2c3c6fd1-c6a4-44e7-8a86-b1f44c1e2572",
    "filename": null,
    "applies_to_exp": null,
    "miq_policy_id": null,
    "notes": null,
    "read_only": null
  }
]
}

```

Find the Policy action to use

Request:

```
GET /api/policy_actions?expand=resources&filter[]=name=compliance_failed
```

Response:

```

{
  "name": "policy_actions",
  "count": 42,
  "subcount": 1,
  "subquery_count": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/policy_actions/34",
      "id": "34",
      "name": "compliance_failed",
      "description": "Mark as Non-Compliant",
      "created_on": "2019-08-08T19:40:10Z",
      "updated_on": "2019-08-08T19:40:10Z",
      "guid": "47969a3f-4eb8-4e7b-b7d2-542e4c0cb0d5",
      "action_type": "default",
      "options": {
      }
    }
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/policy_actions"
    }
  ],
  "links": {

```

```

    "self": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0",
    "first": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0",
    "last": "http://localhost:3000/api/policy_actions?
attributes=name%2Cname&expand=resources&filter%5B%5D=name%3Dcompliance_failed&offset=0"
  }
}

```

Find the Policy event to use

```
GET /api/events?expand=resources&filter[]=name=vm_compliance_failed
```

Response:

```

{
  "name": "events",
  "count": 191,
  "subcount": 1,
  "subquery_count": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/events/119",
      "id": "119",
      "name": "vm_compliance_check",
      "description": "VM Compliance Check",
      "created_on": "2019-08-08T19:40:12Z",
      "updated_on": "2019-08-08T19:40:12Z",
      "guid": "f95fd930-b70d-4198-b018-7f2fe37a33f9",
      "event_type": "Default",
      "definition": null,
      "default": null,
      "enabled": null
    }
  ],
  "actions": [
    {
      "name": "query",
      "method": "post",
      "href": "http://localhost:3000/api/events"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/events?
expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0",
    "first": "http://localhost:3000/api/events?
expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0",
    "last": "http://localhost:3000/api/events?
expand=resources&filter%5B%5D=name%3Dvm_compliance_check&offset=0"
  }
}

```

Create the Compliance Policy for Finance Servers

Request:

```
POST /api/policies
```

```
{
  "name" : "finance_server_policy",
  "description" : "Finance Server Policy",
  "mode" : "compliance",
  "towhat" : "Vm",
  "conditions_ids" : [ "8", "9" ],
  "policy_contents" : [
    {
      "event_id" : "119",
      "actions" : [
        {
          "action_id" : "34",
          "opts" : { "qualifier" : "failure" }
        }
      ]
    }
  ]
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/policies/11",
      "id": "11",
      "name": "finance_server_policy",
      "description": "Finance Server Policy",
      "created_on": "2020-03-25T16:25:06Z",
      "updated_on": "2020-03-25T16:25:06Z",
      "expression": null,
      "towhat": "Vm",
      "guid": "95f28050-7947-41e4-8082-ee9cd512b5fd",
      "created_by": null,
      "updated_by": null,
      "notes": null,
      "active": true,
      "mode": "compliance",
      "read_only": null
    }
  ]
}
```

Create the Compliance Policy Profile for Finance Servers

Request:

POST /api/policy_profiles

```
{
  "name" : "finance_policy_profile",
  "description" : "Finance Servers Policy Profile",
  "mode" : "compliance"
}
```

Response:

```
{
  "results": [
    {
      "href": "http://localhost:3000/api/policy_profiles/35",
```

```

    "id": "35",
    "name": "finance_policy_profile",
    "description": "Finance Servers Policy Profile",
    "set_type": "MiqPolicySet",
    "created_on": "2020-03-24T18:56:23Z",
    "updated_on": "2020-03-24T18:56:23Z",
    "guid": "d4546f30-15a7-48f3-a793-cfd6d1d849a2",
    "read_only": null,
    "set_data": null,
    "mode": "compliance",
    "owner_type": null,
    "owner_id": null,
    "userid": null,
    "group_id": null
  }
]
}

```

Assigning new policy to policy profile

Request:

POST /api/policy_profiles/35/policies

```

{
  "action" : "assign",
  "resources" : [
    { "id" : "11" }
  ]
}

```

Response:

```

{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy: id:'11' description:'Finance Server Policy'
guid: '95f28050-7947-41e4-8082-ee9cd512b5fd'",
      "href": "http://localhost:3000/api/policy_profiles/35",
      "policy_id": "12",
      "policy_href": "http://localhost:3000/api/policies/11"
    }
  ]
}

```

Assigning the policy profile to a virtual machine

Request:

POST /api/vms/171/policy_profiles

```

{
  "action" : "assign",
  "resources" : [
    { "id" : "35" }
  ]
}

```

Response:

```
{
  "results": [
    {
      "success": true,
      "message": "Assigning Policy Profile: id:'35' description:'Finance Servers Policy Profile' guid:'d4546f30-15a7-48f3-a793-cfd6d1d849a2'",
      "href": "http://localhost:3000/api/vms/171",
      "policy_profile_id": "35",
      "policy_profile_href": "http://localhost:3000/api/policy_profiles/35"
    }
  ]
}
```

Querying the policy profiles and related policies of the virtual machine

Request:

```
GET /api/vms/171/policy_profiles?expand=resources&attributes=miq_policies
```

Response:

```
{
  "name": "policy_profiles",
  "count": 6,
  "subcount": 1,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/171/policy_profiles/35",
      "id": "35",
      "name": "finance_policy_profile",
      "description": "Finance Servers Policy Profile",
      "set_type": "MiqPolicySet",
      "created_on": "2020-03-24T18:56:23Z",
      "updated_on": "2020-03-24T18:56:23Z",
      "guid": "d4546f30-15a7-48f3-a793-cfd6d1d849a2",
      "read_only": null,
      "set_data": null,
      "mode": "compliance",
      "owner_type": null,
      "owner_id": null,
      "userid": null,
      "group_id": null,
      "miq_policies": [
        {
          "href": "http://localhost:3000/api/policies/12",
          "id": "12",
          "name": "finance_server_policy",
          "description": "Finance Server Policy",
          "created_on": "2020-03-25T16:44:19Z",
          "updated_on": "2020-03-25T16:44:19Z",
          "expression": null,
          "towhat": "Vm",
          "guid": "e56d468c-1b1b-42f1-a4cb-4ea11924d7b3",
          "created_by": null,
          "updated_by": null,
          "notes": null,
          "active": true,
          "mode": "compliance",
          "read_only": null
        }
      ]
    }
  ]
}
```



```

    }
  ],
  "actions": [
    {
      "name": "assign",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    },
    {
      "name": "unassign",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    },
    {
      "name": "resolve",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171/policy_profiles"
    }
  ],
  "links": {
    "self": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0",
    "first": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0",
    "last": "http://localhost:3000/api/vms/171/policy_profiles?attributes=miq_policies&expand=resources&offset=0"
  }
}

```

At this point, the compliance policy is properly assigned to the virtual machine. If configured via a schedule for the encompassing region, the compliance checks will run as scheduled. They can also be executed on the specific virtual machine via the API as follows:

Requesting a compliance check on the virtual machine

Request:

POST /api/vms/171

```

{
  "action" : "check_compliance"
}

```

Response:

```

{
  "success" : true,
  "message": "VM id:171 name:'aab-ldap' check compliance requested",
  "task_id": "35",
  "task_href": "http://localhost:3000/api/tasks/41",
  "href": "http://localhost:3000/api/vms/171"
}

```

Waiting for the compliance check to complete

One can find out when the compliance check completes by querying the task generated for it and look for the state attribute as follows:

GET /api/tasks/41

Response:

```
{
  "href": "http://localhost:3000/api/tasks/41",
  "id": "41",
  "name": "VM id:171 name:'aab-ldap' check compliance requested",
  "state": "Finished",
  "status": "Ok",
  "message": "Task completed successfully",
  "userid": "admin",
  "created_on": "2020-04-09T21:16:07Z",
  "updated_on": "2020-04-09T21:16:12Z",
  "pct_complete": null,
  "context_data": null,
  "results": null,
  "miq_server_id": "3",
  "identifier": null,
  "started_on": "2020-04-09T21:16:09Z",
  "zone": null,
  "actions": [
    {
      "name": "delete",
      "method": "post",
      "href": "http://localhost:3000/api/tasks/41"
    },
    {
      "name": "delete",
      "method": "delete",
      "href": "http://localhost:3000/api/tasks/41"
    }
  ]
}
```

Querying all compliances run for the virtual machine

Request:

```
GET /api/vms/171/compliances?expand=resources
```

Response:

```
{
  "name": "compliances",
  "count": 430,
  "subcount": 2,
  "pages": 1,
  "resources": [
    {
      "href": "http://localhost:3000/api/vms/171/compliances/40",
      "id": "40",
      "resource_id": "171",
      "resource_type": "VmOrTemplate",
      "compliant": false,
      "timestamp": "2020-03-26T14:30:24Z",
      "updated_on": "2020-03-26T14:30:24Z",
      "event_type": "vm_compliance_check"
    },
    {
      "href": "http://localhost:3000/api/vms/171/compliances/41",
      "id": "41",
      "resource_id": "171",
      "resource_type": "VmOrTemplate",

```

```

        "compliant": false,
        "timestamp": "2020-03-26T20:11:53Z",
        "updated_on": "2020-03-26T20:11:53Z",
        "event_type": "vm_compliance_check"
    }
],
"links": {
    "self": "http://localhost:3000/api/vms/171/compliances?
expand=resources&offset=0",
    "first": "http://localhost:3000/api/vms/171/compliances?
expand=resources&offset=0",
    "last": "http://localhost:3000/api/vms/171/compliances?
expand=resources&offset=0"
}
}

```

Querying the virtual machine and its latest compliance result including the last compliance status

Request:

```
GET /api/vms/171?attributes=last_compliance,last_compliance_status
```

Response:

```

{
    "href": "http://localhost:3000/api/vms/171",
    "id": "171",
    "vendor": "vmware",
    "format": null,
    "version": null,
    "name": "aab-ldap",
    "description": null,
    "location": "aab-ldap/aab-ldap.vmx",
    "config_xml": null,
    "autostart": null,
    "host_id": 2,
    "last_sync_on": null,
    "created_on": "2020-03-10T18:53:36Z",
    "updated_on": "2020-03-26T01:58:16Z",
    "storage_id": "7",
    "guid": "30be2fa7-6b95-4a62-a163-d3107fc79d94",
    "ems_id": "2",
    "last_scan_on": "2020-03-18T21:13:32Z",
    "last_scan_attempt_on": "2020-03-18T21:12:25Z",
    "uid_ems": "4cda25de-c41c-4f68-8b3d-0cca0dc993c4",
    "retires_on": null,
    "retired": null,
    "boot_time": null,
    "tools_status": null,
    "standby_action": null,
    "power_state": "on",
    "state_changed_on": "2020-03-10T18:53:36Z",
    "previous_state": null,
    "connection_state": "connected",
    "last_perf_capture_on": "2020-03-26T01:58:16Z",
    "registered": null,
    "busy": null,
    "smart": null,
    "memory_reserve": null,
    "memory_reserve_expand": null,

```

```

"memory_limit": null,
"memory_shares": null,
"memory_shares_level": null,
"cpu_reserve": null,
"cpu_reserve_expand": null,
"cpu_limit": -1,
"cpu_shares": 200,
"cpu_shares_level": "normal",
"cpu_affinity": null,
"ems_created_on": null,
"template": false,
"evm_owner_id": null,
"miq_group_id": "1",
"linked_clone": true,
"fault_tolerance": null,
"type": "ManageIQ::Providers::Vmware::InfraManager::Vm",
"ems_ref": "vm-825",
"ems_cluster_id": "3",
"retirement_warn": null,
"retirement_last_warn": null,
"vnc_port": null,
"flavor_id": "960",
"availability_zone_id": "23",
"cloud": true,
"retirement_state": null,
"cloud_network_id": null,
"cloud_subnet_id": null,
"cloud_tenant_id": "2",
"raw_power_state": "ACTIVE",
"publicly_available": null,
"orchestration_stack_id": null,
"retirement_requester": null,
"tenant_id": "1",
"resource_group_id": null,
"deprecated": null,
"storage_profile_id": null,
"cpu_hot_add_enabled": null,
"cpu_hot_remove_enabled": false,
"memory_hot_add_enabled": true,
"memory_hot_add_limit": null,
"memory_hot_add_increment": null,
"hostname": null,
"ems_ref_type": "VirtualMachine",
"restart_needed": null,
"last_compliance_status": false,
"last_compliance": {
  "id": "41",
  "resource_id": "1000000132824",
  "resource_type": "VmOrTemplate",
  "compliant": false,
  "timestamp": "2020-03-26T20:11:53Z",
  "updated_on": "2020-03-26T20:11:53Z",
  "event_type": "vm_compliance_check"
},
"actions": [
  {
    "name": "edit",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "add_event",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  }
]

```

```

    },
    {
      "name": "refresh",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shutdown_guest",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "reboot_guest",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "start",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "stop",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "suspend",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shelve",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "shelve_offload",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "pause",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "request_console",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "reset",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "request_retire",
      "method": "post",
      "href": "http://localhost:3000/api/vms/171"
    },
    {
      "name": "delete",
      "method": "post",

```

```

    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "set_owner",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "set_ownership",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "set_miq_server",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "scan",
    "method": "post",
    "href": "http://localhost:3000/api/vms/171"
  },
  {
    "name": "delete",
    "method": "delete",
    "href": "http://localhost:3000/api/vms/171"
  }
]
}

```

Managed services APIs

This section includes the prerequisites to use Managed services APIs, Common Managed services APIs, sample Python script, and API reference.

- [API reference](#)
- [Prerequisites](#)
- [Common APIs](#)
- [Sample Python script](#)
- [Content APIs](#)
- [Retrieve deployed APIs](#)

API reference

You can use REST APIs to integrate Managed services with your application.

About the task

The terms in the swagger files for your reference:

- Stack is the API name for a deployed template instance.

- Stackstate is a separate table that keeps a history of all the detailed states the stack has been in. When you do a `stack/<stack_id>/retrieve`, it includes the latest state.
- Foreign key (fk) is the ID of that particular stackstate.

Procedure

1. Before you run API commands, generate access tokens. For more information about the steps, see [Generating access tokens](#).
2. Copy the following Swagger files and save them in yaml format:
 - [Managed services: cam-iaas](#)
The Managed services IaaS component provides support for defining connections to various clouds, templates that can be used to deploy stack instances.
 - [Managed services: cam-service-composer-api](#)
The Managed services service composer.
 - [Managed services: cam-tenant-api](#)
The Managed services tenant service provides a way to retrieve information about the tenant for the current user.
3. Managed services API needs tenantId, IBM Cloud Pak for AIOps team (ace_orgGuid), and IBM Cloud Pak for AIOps namespace (cloudOE_spaceGuid) for authorization. For the procedure to get these values, see [Prerequisites for using Managed services APIs](#) topic.
4. Load the yaml files onto any swagger browser.

Prerequisites for using Managed services APIs

This section includes prerequisite steps to use Managed services APIs.

The following table describes the variables that are used throughout the API calls.

Variable	Description
TSA_HOSTNAME	TSA_HOSTNAME is the HOST/PORT value in the output of the command: <code>oc get routes cam -o=jsonpath='{.spec.host}'</code> .
TSA_USER	TSA_USER is the IBM Cloud Pak for AIOps user you want to use. You can use either the default admin user or any user you have added/configured in the IBM Cloud Pak for AIOps.
TSA_USER_PASSWORD	TSA_USER_PASSWORD is the password for the TSA_USER .
MCM_CLUSTER_HOSTNAME	MCM_CLUSTER_HOSTNAME is the cluster hostname for IBM Cloud Pak for AIOps. It is the HOST/PORT value in the output of the command: <code>\$ oc -n <IA_installation_namespace> get route cp-console</code>

Where **<IA_installation_namespace>** is the namespace that Infrastructure Automation is installed in.

1. Preparing to run Managed services APIs:

- Set the following environment variables:
 - Run `vi setenv` to create a script.
 - In `setenv`, paste the following environment variables:
 - `export TSA_HOSTNAME=<Replace_with_your_TSA_HOSTNAME>`
 - `export TSA_USER=<Replace_with_your_CloudPak_for_MCM_User>`
 - `export TSA_USER_PASSWORD=<Replace_with_your_TSA_ADMIN_USER_PASSWORD>`
 - `export MCM_CLUSTER_HOSTNAME=<Replace_with_your_MCM_CLUSTER_HOSTNAME>`
 - Run `chmod +x setenv` to execute permissions.
 - Run `./setenv` script within the context of your current session.
- To get an access token to your API calls and authenticate your credentials, see [Get access token by using your username and password](#).
- Most Managed services APIs require `tenantId` and IBM Cloud Pak for AIOps `team (ace_orgGuid)` as query parameters. Managed services APIs related to deployed template or service instances also require `namespace (cloudOE_spaceGuid)` as a query parameter. The "Admin" user has access to all teams and namespaces.

To determine the Managed services `tenantId`, IBM Cloud Pak for AIOps `team`, and `namespace` issue the following API by using the access token from the previous API:

```
curl -k -X GET -H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
https://${TSA_HOSTNAME}/cam/tenant/api/v1/tenants/getTenantOnPrem
```

- For `tenantId`, save the "id" value from the output:

```
export TSA_TENANT_ID=<response.id>
```

- For IBM Cloud Pak for AIOps `team (ace_orgGuid)`, save the "teamId" from the namespace you would like to use:

```
export MCM_TEAM=<response.namespaces[0].teamId>
```

- For `namespace (cloudOE_spaceGuid)`, save the "uid" from the namespace you would like to use:

```
export NAMESPACE=<response.namespaces[0].uid>
```

The following examples of `getTenantOnPrem` API are for "Admin" user and LDAP user:

- When you call `https://${TSA_HOSTNAME}/cam/tenant/api/v1/tenants/getTenantOnPrem` with the admin account

```
{
  "id": "1ed535a5-c895-4e5c-85fe-d8403a6c5331", <-- tenantId
  "name": "ICP Account",
  "ownerId": "On_Prem",
  "wlpClientId": "XRw1JM5pQ0scGpwTNzN2gA3dTl1xDt07",
  "namespaces": [
    {
      "name": "default",
      "uid": "default", <-- namespace (cloudOE_spaceGuid)
      "teamId": "all", <-- MCM team (ace_orgGuid)
      "teamName": "all",
      "roles": [
        {
          "id": "crn:v1:icp:private:iam:::role:Administrator"
```



```

    }
  ],
},
{
  "name": "kube-public",
  "uid": "kube-public",
  "teamId": "all",
  "teamName": "all",
  "roles": [
    {
      "id": "crn:v1:icp:private:iam::::role:Administrator"
    }
  ]
},
{
  "name": "kube-system",
  "uid": "kube-system",
  "teamId": "all",
  "teamName": "all",
  "roles": [
    {
      "id": "crn:v1:icp:private:iam::::role:Administrator"
    }
  ]
},
{
  "name": "services",
  "uid": "services",
  "teamId": "all",
  "teamName": "all",
  "roles": [
    {
      "id": "crn:v1:icp:private:iam::::role:Administrator"
    }
  ]
}
]
}

```

- For the "admin" user example, the following output must be saved:
 - `tenantId = response.id = 1ed535a5-c895-4e5c-85fe-d8403a6c5331`
`export TSA_TENANT_ID=1ed535a5-c895-4e5c-85fe-d8403a6c5331`
 - IBM Cloud Pak for AIOps team (ace_orgGuid) =
`response.namespaces[0].teamId = all`
`export MCM_TEAM=all`
Note: The admin does not belong to a team so use the value "all".
 - IBM Cloud Pak for AIOps namespace (cloudOE_spaceGuid)=
`response.namespaces[0].uid = default`
`export NAMESPACE=default`
- When calling
`https://${TSA_HOSTNAME}/cam/tenant/api/v1/tenants/getTenantOnPrem`
 with an LDAP user who is assigned to team "camusers" with access to the "default" namespace.

```

{
  "id": "1ed535a5-c895-4e5c-85fe-d8403a6c5331", <-- tenantId
  "name": "ICP Account",
  "ownerId": "On_Prem",
  "wlpClientId": "kSNhTWdjPWGVhQk8RdHcuNVl19uOCkrf",
  "namespaces": [
    {
      "name": "camdefault - default",
      "uid": "default", <-- namespace
    }
  ]
}

```

For the LDAP user example, the following output can be saved:

- `tenantId = response.id = 1ed535a5-c895-4e5c-85fe-d8403a6c5331`
`export TSA_TENANT_ID=1ed535a5-c895-4e5c-85fe-d8403a6c5331`
- IBM Cloud Pak for AIOps team (`ace_orgGuid`) =
`response.namespaces[0].teamId = camdefault`
`export MCM_TEAM=camdefault`
- namespace (`cloudOE_spaceGuid`)= `response.namespaces[0].uid = default`
`export NAMESPACE=default`

Common Managed services APIs

This section includes common Managed services APIs with examples.

- [Provider and connection APIs](#)
- [Template and template instance APIs](#)
- [Terraform version APIs](#)
- [Terraform file APIs](#)
- [Resource specific APIs](#)
- [Service and service instance APIs](#)
- [Dynamic parameter discovery APIs](#)
- [Application APIs](#)
- [Shared parameter APIs](#)
- [SMTP connection APIS](#)
- [Usage specific APIs](#)
- [Example: Retrieve deployed templates list](#)

Provider and connection APIs

This section includes all provider and connection related APIs with examples.

- [Accessing APIs](#)
- [Retrieve the list of providers](#)
- [Create a cloud connection](#)
- [Retrieve the list of cloud connections](#)
- [Retrieve a specific cloud connection](#)
- [Update a specific cloud connection](#)
- [Delete a specific cloud connection](#)

Accessing APIs

To get an access token to your API calls and authenticate your credentials, see [Accessing APIs](#).

Retrieve the list of providers

Retrieve the list of providers.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/providers?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Example response:

```
[
  {
    "configuration_parameters": [],
    "connection_parameters": [
      {
        "name": "ibmcloud_api_key",
        "type": "password",
        "description": "The API key for IBM Cloud",
        "required": "true",
        "secured": "true",
        "label": "IBM Cloud API Key",
        "value": ""
      },
      {
        "name": "iaas_classic_username",
        "type": "string",
        "description": "The User Name for IBM Cloud Classic
Infrastructure",
        "required": "true",
        "secured": "false",
        "label": "IBM Cloud Classic Infrastructure User Name",
      },
      {
        "name": "iaas_classic_api_key",
        "type": "password",
        "description": "The API key for IBM Cloud Classic Infrastructure",
        "required": "true",

```

```

        "secured": "true",
        "label": "IBM Cloud Classic Infrastructure API Key",
        "value": ""
    },
    ],
    "name": "IBM",
    "description": "IBM Provider",
    "datatype_identifier": "IBM",
    "id": "5c588624df7d9e00187628c7",
    "label": "IBM"
},
...

```

Create a cloud connection

POST to create a cloud connection.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json \
"https://${TSA_HOSTNAME}/cam/api/v1/cloudconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

cc_create.json

```

{
  "name": "AWSTestConnection",
  "description": "My AWS Test Connection",
  "connection_parameters": [
    {
      "name": "aws_access_key_id",
      "type": "string",
      "description": "The API Access Key ID for AWS generated in the Identity and
Access Management (IAM) service",
      "required": "true",
      "secured": "true",
      "label": "Access Key ID",
      "default": null,
      "options": null,
      "value": "1234"
    },
    {
      "name": "aws_secret_key",
      "type": "password",
      "description": "The API Secret Access Key for AWS generated in the Identity
and Access Management (IAM) service",
      "required": "true",
      "secured": "true",
      "label": "Secret Access Key",
      "default": null,
      "options": null,
      "value": "5678"
    },
    {
      "name": "aws_default_region",
      "type": "string",
      "description": "Region to deploy the service on.",
      "required": "true",
      "secured": "false",
      "label": "AWS Region",

```

```

"default": null,
"options": [
  {
    "label": "US East (Northern Virginia)",
    "value": "us-east-1",
    "default": "true"
  },
  {
    "label": "US East (Ohio) Region",
    "value": "us-east-2"
  },
  {
    "label": "US West (Northern California)",
    "value": "us-west-1"
  },
  {
    "label": "US West (Oregon) Region",
    "value": "us-west-2"
  },
  {
    "label": "Asia Pacific (Seoul) Region",
    "value": "ap-northeast-2"
  },
  {
    "label": "Asia Pacific (Singapore) Region",
    "value": "ap-southeast-1"
  },
  {
    "label": "Asia Pacific (Sydney) Region",
    "value": "ap-southeast-2"
  },
  {
    "label": "Asia Pacific (Tokyo) Region",
    "value": "ap-northeast-1"
  },
  {
    "label": "EU (Frankfurt) Region",
    "value": "eu-central-1"
  },
  {
    "label": "EU (Ireland) Region",
    "value": "eu-west-1"
  },
  {
    "label": "EU (London) Region",
    "value": "eu-west-2"
  },
  {
    "label": "AWS GovCloud (US)",
    "value": "us-gov-west-1"
  }
],
"value": "us-east-1"
}
],
"configuration_parameters": [
],
"providerId": "5bb76eb78e04ec001c19466f",
"additional_parameters": [
  {
    "plugin": "plugin-terraform",
    "supported_api_types": [
      "stack",
      "iaas"
    ]
  }
]

```

```

    ],
    "supported_template_types": [
      {
        "template_type": "Terraform",
        "template_formats": [
          "JSON",
          "HCL"
        ]
      }
    ],
    "configuration_parameters": [
    ]
  }
]
}

```

- Provider Id - Using the [Retrieve the list of providers](#) GET API call, retrieve the Provider id. In this example, AWS provider is used.
- `cloudOE_spaceGuid` query parameter governs the namespace to be used when a cloud connection is created. In this example, the cloud connection gets created in the namespace that `$NAMESPACE` resolves to. Now, if you want to create the cloud connection to be globally available and not tied to any specific namespace, then omit the `cloudOE_spaceGuid` query parameter.

GET the list of cloud connections

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/cloudconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Example output:

```

[
  {
    "name": "MyVMware",
    "description": "My VMware vSphere Connection",
    "connection_parameters": [
      {
        "name": "vsphere_user",
        "value": "username@vsphere.local",
        "secured": "false"
      },
      {
        "name": "vsphere_password",
        "value": "*****",
        "secured": "true"
      },
      {
        "name": "vsphere_server",
        "value": "1.2.3.4",
        "secured": "false"
      }
    ],
    "validate": true,
    "message": "",
    "id": "59f3b11bd1b007002263c5f3",
    "providerId": "59f38667d1b007002263c578",
    "additional_Parameters": {
      "0": {
        "plugin": "plugin-terraform"
      }
    }
  }
]

```

```

    },
    "tenantId": "5c0e0a5e-d431-468c-9a11-a3df3978e424"
  }
]

```

The `cloudOE_spaceGuid` query parameter is optional and the search results display all cloud connections that `$NAMESPACE` resolves to.

GET a specific cloud connection (filter)

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/cloudconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&filter=%7B%22where%22%3A%20%7B%2
2name%22%3A%20%22vmware.octravis%22%20%7D%7D"

```

Example output:

```

[
  {
    "name": "vmware.octravis",
    "description": "vmware.octravis",
    "connection_parameters": [
      {
        "name": "vsphere_user",
        "value": "octravis@vsphere.local",
        "secured": "false"
      },
      {
        "name": "vsphere_password",
        "value": "*****",
        "secured": "true"
      },
      {
        "name": "vsphere_server",
        "value": "1.2.3.123",
        "secured": "false"
      }
    ],
    "configuration_parameters": [
      {
        "name": "",
        "value": ""
      }
    ],
    "validate": true,
    "message": "",
    "id": "5a4d094a47ff9c0022728759",
    "providerId": "5a4cf14b47ff9c0022728676",
    "additional_Parameters": [
      {
        "plugin": "plugin-terraform"
      }
    ],
    "tenantId": "15e7d67e-9f73-43cd-9759-b39ebaea85e8"
  }
]

```

Update a specific cloud connection

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_update.json \
"https://${TSA_HOSTNAME}/cam/api/v1/cloudconnections/${CAM_CLOUD_CONNECTION_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example request body:

```
{
  "name": "Updated connection name",
  "description": "Updated connection description"
}
```

Delete a specific cloud connection

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/cloudconnections/${CAM_CLOUD_CONNECTION_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Template and template instance APIs

This section includes template and template instance APIs with examples.

- [Create a template from a file](#)
- [Create a template from a source](#)
- [Duplicate a template](#)
- [Retrieve the list of templates](#)
- [Retrieve a specific template using filter](#)
- [Retrieve a specific template by id](#)
- [Create new template version for an existing template](#)
- [Deploy a template](#)
- [Retrieve the list of deployed templates](#)
- [Retrieve details for a template instance](#)
- [Modify a template instance](#)
- [Apply the changes to a template instance](#)
- [Destroy a template instance](#)
- [Delete a template instance \(stack\) after destroying it](#)
- [Retrieve stack information](#)
- [Download the Terraform state \(tfstate\) for a template instance](#)
- [Download the Terraform state \(tfstate\) for a specific stack state](#)
- [Upload the Terraform state \(tfstate\) for a template instance](#)
- [Download the execution plan for a template instance](#)
- [Retrieve the execution plan in json format for a template instance](#)

Create a template from a file

```
curl -k -X POST \
-H "Authorization: ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
```



```

-d @template_create.json \
"https://${TSA_HOSTNAME}/cam/api/v1/templates?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

template_create.json:
{
  "name": "mytemplate",
  "description": "here is the description",
  "type": "userCreated",
  "manifest": {
    "template_type": "Terraform",
    "template": {
      "templateData": "variable \"region\" {\n  default = \"us-east-
1\"\n}\n",
      "templateOutput": "",
      "templateVariables": {
        "template_input_params": [
          {
            "name": "region",
            "description": "variable description",
            "default": "us-east-1",
            "type": "string",
            "required": true,
            "secured": false,
            "label": "region",
            "hidden": false,
            "immutable": false
          }
        ]
      }
    }
  },
  "template_provider": "IBM"
},
"metadata": {
  "displayName": "the template display name"
}
}

```

The `cloudOE_spaceGuid` query parameter governs the namespace to be used when a template is created. In this example, the template gets created in the namespace that `$NAMESPACE` resolves to. Now, if you want to create the template to be globally available and not tied to any specific namespace, then omit the `cloudOE_spaceGuid` query parameter.

Example output:

```

{
  "description": "here is the description",
  "name": "mytemplate",
  "type": "userCreated",
  "manifest": {
    "template_type": "Terraform",
    "template_provider": "IBM",
    "template": {
      "templateData": "variable \"region\" {\n  default = \"us-east-
1\"\n}\n",
      "templateVariables": {
        "template_input_params": [
          {
            "name": "region",
            "description": "variable description",
            "default": "us-east-1",
            "type": "string",
            "required": true,

```

```

        "secured": false,
        "label": "region",
        "hidden": false,
        "immutable": false
      }
    ]
  },
  "metadata": {
    "displayName": "the template display name"
  },
  "created_at": "2018-01-09T05:10:48.523Z",
  "id": "5a544ed88bec4000200dc328",
  "tenantId": "cd3ed377-8167-4078-b486-309b1f237928",
  "default_template_version": "5a544ed88bec4000200dc329",
  "template_version": {
    "id": "5a544ed88bec4000200dc329",
    "name": "v1.0.0",
    "enabled": true
  },
  "template_versions": [
    {
      "id": "5a544ed88bec4000200dc329",
      "name": "v1.0.0",
      "enabled": true
    }
  ]
}

```

Create a template from a source

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @template_input.json \
"https://${TSA_HOSTNAME}/cam/api/v1/templates/createFromSource?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

template_input.json:
{
  "template_type": "Terraform",
  "template_format": "HCL",
  "template_source": {
    "github": {
      "url": "https://github.com/org/provider-terraform-base",
      "token": "access token for github repository",
      "dir": "subdir/testTemplate",
      "ref": "master"
    }
  }
}

```

The `cloudOE_spaceGuid` query parameter governs the namespace to be used when a template is created. In this example, the template gets created in the namespace that `$NAMESPACE` resolves to. Now, if you want to create the template to be globally available and not tied to any specific namespace, then omit the `cloudOE_spaceGuid` query parameter.

Example output:

```

{
  "description": "Template contents are hosted in a git repository.",
  "name": "New template hosted in git 129c26100b5b9fe7",
  "type": "userCreated",
  "manifest": {
    "template_type": "Terraform",
    "template_source": {
      "github": {
        "url": "https://github.com/org/provider-terraform-base",
        "token": "1234567890",
        "dir": "subdir/testTemplate1",
        "ref": "master"
      },
      "displayUrl": "https://github.com/org/provider-terraform-
base/tree/master/subdir/testTemplate1",
      "displayUrlReplaceRef": "https://github.com/org/provider-terraform-
base/tree/REPLACE_REF/subdir/testTemplate1"
    },
    "template": {
      "templateVariables": {}
    }
  },
  "metadata": {
    "displayName": "",
    "longDescription": ""
  },
  "created_at": "2018-01-09T05:24:30.288Z",
  "id": "5a54520e8bec4000200dc32a",
  "tenantId": "cd3ed377-8167-4078-b486-309b1f237928",
  "default_template_version": "5a54520e8bec4000200dc32b",
  "template_version": {
    "id": "5a54520e8bec4000200dc32b",
    "ref": "master",
    "name": "master",
    "enabled": true,
    "displayUrl": "https://github.com/org/provider-terraform-
base/tree/master/subdir/testTemplate1"
  },
  "template_versions": [
    {
      "id": "5a54520e8bec4000200dc32b",
      "ref": "master",
      "name": "master",
      "enabled": true,
      "displayUrl": "https://github.com/org/provider-terraform-
base/tree/master/subdir/testTemplate1"
    }
  ]
}

```

Example data to import a template whenever you use a compressed file:

```

{
  "template_type": "Terraform",
  "template_source": {
    "archive": {
      "url": "https://myserver.myorg.com/templates/template1.zip",
    }
  }
}

```

The `artifactory_api_key` is an optional parameter that allows you to specify an API key to an artifactory location. If the file is hosted in artifactory and an API key is needed, then use the `artifactory_api_key`.

For example:

```
{
  "template_type": "Terraform",
  "template_source": {
    "archive": {
      "url": "https://myserver.myorg.com/templates/template1.zip",
      "artifactory_api_key": "YOURKEYHERE"
    }
  }
}
```

If you get an error during the API call, ensure that your archive is in "zip" format and your file is downloadable through HTTP/HTTPS.

Duplicate a template

This API duplicates the template along with all versions.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @duplicate_template.json \
"https://${TSA_HOSTNAME}/cam/api/v1/templates/${CAM_TEMPLATE_ID}/duplicate?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

The `cloudOE_spaceGuid` query parameter governs the namespace to be used when the duplicate template is created. In this example, the template gets created in the namespace that `$NAMESPACE` resolves to. Now, if you want to create the template to be globally available and not tied to any specific namespace, then omit the `cloudOE_spaceGuid` query parameter.

Example request body:

```
{
  "name": "mynewname"
}
```

Retrieve the list of templates

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/templates?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

The `cloudOE_spaceGuid` query parameter is optional and the search results display all templates that are created for the namespace `$NAMESPACE`. The result also includes all the templates that are globally available.

Example output:

```
[
  {
    "description": "Kubernetes - An orchestration system for containerized
applications, e.g., Nginx.",
    "name": "Kubernetes Cluster with Nginx Deployment",
    "type": "prebuilt",
    "manifest": {
      "template_type": "Terraform",
      "template_format": "HCL",
```

```

        "template_provider": "VMware vSphere"
    },
    "metadata": {
        "displayName": "Kubernetes Cluster with Nginx Deployment",
        "longDescription": "Deploys a Kubernetes Cluster in VMware, where
Kubernetes is an open-source system for automating deployment, scaling, and
management of containerized applications. This deployment also deploys the NGINX
sample application to validate the deployment is operational, where NGINX is an
open source web server and load balancer for high-traffic sites.",
        "bullets": [
            {
                "title": "Clouds",
                "description": "VMware"
            }
        ]
    },
    "created_at": "2017-10-27T19:17:59.874Z",
    "id": "59f38667d1b007002263c581",
    "tenantId": "DEFAULT",
    "default_template_version": "59f38667d1b007002263c582"
},

```

Retrieve a specific template using filter

This API returns the template metadata along with all the template versions.

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://$TSA_HOSTNAME/cam/api/v1/templates?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&filter=%7B%22where%22%3A%20%7B%2
2name%22%3A%20%22SingleVirtualMachine%22%20%7D%7D"

```

Example output:

```

[
  {
    "description": "Create Virtual Machine with single vnic",
    "name": "SingleVirtualMachine",
    "type": "prebuilt",
    "manifest": {
      "template_type": "Terraform",
      "template_format": "HCL",
      "template_provider": "VMware vSphere"
    },
    "metadata": {
      "displayName": "SingleVirtualMachine",
      "longDescription": "Create Virtual Machine with single vnic on a
network label by cloning"
    },
    "created_at": "2017-11-03T15:19:30.566Z",
    "id": "59fc89024722a1001ff2974c",
    "tenantId": "DEFAULT",
    "default_template_version": "59fc89024722a1001ff2974d"
  }
]

```

Retrieve a specific template by id

This API returns the template metadata along with all the template versions.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/templates/${CAM_TEMPLATE_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Create new template version for an existing template

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @template_version.json \
"https://${TSA_HOSTNAME}/cam/api/v1/templateVersions?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Examples of input template_version.json:

- Create a new template version from git:

```
{
  "name": "v2.0",
  "ref": "master",
  "templateId": "parent template id"
}
```

- Create a template version from file:

```
{
  "name": "v2.0",
  "templateId": "parent template id",
  "templateData": "variable \"region\" {\n  default = \"us-east-1\"\n}\n",
  "templateVariables": {
    "template_input_params": [
      {
        "name": "region",
        "description": "variable description",
        "default": "us-east-1",
        "type": "string",
        "required": true,
        "secured": false,
        "label": "region",
        "hidden": false,
        "immutable": false
      }
    ]
  }
}
```

Deploy a template

Get the template versions using the [Retrieve a specific template by id](#) if you want to specify a particular template version while deploying the template.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @template_input.json \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example input:

```
template_input.json:
{
  "name": "vSphere95",
  "cloud_connection_ids": [
    "5a00ab148af3f3002010ab45"
  ],
  "parameters": [
    {
      "name": "datacenter",
      "value": "Rack-8283_1000167"
    },
    {
      ....
    },
  ],
}
"templateId": "5a00a2448af3f3002010aae1",
"templateVersionId": "5a00a2448af3f3002010aae2",
"catalogName": "SingleVirtualMachine",
"catalogType": "userCreated",
"forceCreate": "true"
}
```

Note: `templateVersionId` is an optional parameter, if it is not provided, the default provided template version is used for deployment.

When the template to be deployed has `input_datatypes`, then use the following information:

Pass the input dataobjects that you want to use as your input to a stack by using the name and Data Type. It uniquely identifies the dataobject instance you want to use. For example:

```
"input_dataobjects": [
  {
    "name": "DefaultNoBastionHostRequired",
    "datatype": "bastionhost"
  }
],
```

Where:

- **bastionhost** is the Data Type name. It matches the datatype in the `input_datatypes` of the template you are deploying.
- **name** is the name of the Data Object instance you want to use when you deploy this instance.

For each `input_datatype`, the template requires an `input_dataobject`.

In this example, the template requires an `input_datatype` of "bastionhost". When you deploy a stack from this template, provide the dataobject to use that implements the "bastionhost" datatype:

```
"input_dataobjects": [
  {
    "name": "DefaultNoBastionHostRequired",
    "datatype": "bastionhost"
  }
],
```

Example Response:

```
{
  "name": "vSphere95",
  "created_at": "2017-11-06T20:18:11.697Z",
  "templateId": "5a00a2448af3f3002010aae1",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "prebuilt",
  "templateProvider": "VMware vSphere",
  "template_type": "Terraform",
  "template_format": "HCL",
  "applied_state_id": "5a00c3838af3f3002010ab53",
  "latest_state_id": "5a00c3838af3f3002010ab53",
  "cloud_connection_ids": [
    "5a00ab148af3f3002010ab45"
  ],
  "id": "5a00c3838af3f3002010ab53",
  "tenantId": "df87bf8a-2ea9-478d-9b37-2b36da4af6ed",
  "status": "IN_PROGRESS",
  "message": "",
  "action": "CREATE",
  "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
  "applied_status": "IN_PROGRESS",
  "applied_action": "CREATE",
  "data": {
    "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
    "stack_job_id": "5a00c3838af3f3002010ab53",
    "stack_job_is_latest": true,
    "action": "CREATE",
    "status": "IN_PROGRESS",
    "tenantId": "df87bf8a-2ea9-478d-9b37-2b36da4af6ed"
  }
}
```

Create a stack (advanced scenario)

Use this procedure if you want to do a plan on the stack before you deploy:

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @template_input.json \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example input:

```
{
  "name": "vSphere95",
  "cloud_connection_ids": [
    "5a00ab148af3f3002010ab45"
  ],
  "parameters": [
    {
      "name": "datacenter",
      "value": "Rack-8283_1000167"
    },
    {
      ....
    },
  ],
  "templateId": "5a00a2448af3f3002010aae1",
```



```
"templateVersionId": "5a00a2448af3f3002010aae2",
"catalogName": "SingleVirtualMachine",
"catalogType": "userCreated",
"forcePlan": "true"
}
```

Retrieve the list of deployed templates

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example output:

```
[
  {
    "name": "My WAS Liberty",
    "created_at": "2017-10-29T16:03:20.985Z",
    "templateId": "59f38692d1b007002263c5e3",
    "namespaceId": "default",
    "templateName": "IBM WebSphere Application Server Liberty V17 on a single
virtual machine - VMware vSphere",
    "templateCategory": "chefcontent",
    "templateProvider": "VMware vSphere",
    "template_type": "Terraform",
    "template_format": "HCL",
    "applied_state_id": "59f5fbc9d1b007002263c5fa",
    "latest_state_id": "59f5fbc9d1b007002263c5fa",
    "cloud_connection_ids": [
      "59f3b11bd1b007002263c5f3"
    ],
    "id": "59f5fbc8d1b007002263c5f9",
    "tenantId": "5c0e0a5e-d431-468c-9a11-a3df3978e424",
    "status": "SUCCESS",
    "message": "",
    "action": "CREATE",
    "stack_id": "afa51b40-bcc2-11e7-bbfe-b56e007b5b00",
    "applied_status": "SUCCESS",
    "applied_action": "CREATE",
    "applied_output_namespace": "content_template_output"
  }
]
```

The `cloudOE_spaceGuid` query parameter is optional and the search results display all template instances that `$NAMESPACE` resolves to.

Retrieve details for a template instance

Example:

```
export STACK_ID=5a00c3838af3f3002010ab53
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${STACK_ID}/retrieve?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example response:

```
{
  "name": "vSphere95",
  "created_at": "2017-11-06T20:18:11.697Z",
  "templateId": "5a00a2448af3f3002010aae1",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "prebuilt",
  "templateProvider": "VMware vSphere",
  "template_type": "Terraform",
  "template_format": "HCL",
  "applied_state_id": "5a00c3838af3f3002010ab54",
  "latest_state_id": "5a00c3838af3f3002010ab54",
  "cloud_connection_ids": [
    "5a00ab148af3f3002010ab45"
  ],
  "id": "5a00c3838af3f3002010ab53",
  "tenantId": "df87bf8a-2ea9-478d-9b37-2b36da4af6ed",
  "status": "SUCCESS",
  "message": "",
  "action": "CREATE",
  "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
  "applied_status": "SUCCESS",
  "applied_action": "CREATE",
  "data": {
    "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
    "stack_job_id": "5a00c3834e13b5001f4aecdl",
    "stack_job_is_latest": true,
    "status": "SUCCESS",
    "action": "CREATE",
    "message": "\nInitializing provider plugins...\nTerraform has been
successfully initialized!\n\nYou may now begin working with Terraform. Try running
\"terraform plan\" to see\nany changes that are required for your infrastructure.
All Terraform commands\nshould now work.\n\nIf you ever set or change modules or
backend configuration for Terraform,\nrerun this command to reinitialize your
working directory. If you forget, other\ncommands will detect it and remind you to
do so if necessary.\ndata.vsphere_datacenter.datacenter: Refreshing
state...\nvsphere virtual machine.vm 1: Creating...\n cluster:
```

Modify a template instance (stack)

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @plan_input.json \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${STACK_ID}/plan?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

plan_input.json:
{
  "templateVersionId": "59fc89024722a1001ff2974d",
  "parameters": [
    {
      "name": "datacenter",
      "value": "Rack-8283_1000167"
    },
    {
      ...
    },
  ],
}
```

For the plan request body, the parameters are optional and the execution plan gets created by using the parameter values from the initial deploy.

Example response:

```
{
  "name": "vSphere95",
  "created_at": "2017-12-13T14:48:22.937Z",
  "templateId": "5a2b00d7b17b800020af2e7c",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "prebuilt",
  "templateProvider": "VMware vSphere",
  "template_type": "Terraform",
  "applied_state_id": "5a313db7b17b800020af300a",
  "latest_state_id": "5a314639b17b800020af3023",
  "cloud_connection_ids": [
    "5a2b0513b17b800020af2f2b"
  ],
  "id": "5a313db6b17b800020af3009",
  "tenantId": "a8fbbe67-1480-49c0-be60-0084b31a0942",
  "status": "IN_PROGRESS",
  "message": "",
  "action": "PLAN",
  "stack_id": "ab100110-e014-11e7-85f3-c3bc8b9fdda1",
  "applied_status": "SUCCESS",
  "applied_action": "CREATE",
  "data": {
    "stack_id": "ab100110-e014-11e7-85f3-c3bc8b9fdda1",
    "stack_job_id": "5a3146396e53230020466164",
    "stack_job_is_latest": true,
    "action": "PLAN",
    "status": "IN_PROGRESS",
    "tenantId": "a8fbbe67-1480-49c0-be60-0084b31a0942"
  }
}
```

A successful plan has two states, namely SUCCESS AND SUCCESS_WITH_CHANGES:

- The SUCCESS indicates that the plan is successful and there will not be any changes to your environment even if you apply the plan.
- The SUCCESS_WITH_CHANGES indicates that the plan is successful, but when you apply it, there will be changes to your environment. Go through the logs to understand more about the changes to the environment. For example, a template is defined to create a virtual machine. You make modifications to the template such that it creates a second virtual machine. The resultant SUCCESS_WITH_CHANGES indicates that if you apply the plan, the second virtual machine will get created.

Apply the changes to a template instance (stack)

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${STACK_ID}/apply?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example output:

```
{
  "name": "vSphere95",
  "created_at": "2017-12-13T14:48:22.937Z",
```

```

    "templateId": "5a2b00d7b17b800020af2e7c",
    "namespaceId": "default",
    "templateName": "SingleVirtualMachine",
    "templateCategory": "prebuilt",
    "templateProvider": "VMware vSphere",
    "template_type": "Terraform",
    "applied_state_id": "5a314892b17b800020af3042",
    "latest_state_id": "5a314892b17b800020af3042",
    "cloud_connection_ids": [
      "5a2b0513b17b800020af2f2b"
    ],
    "id": "5a313db6b17b800020af3009",
    "tenantId": "a8fbbe67-1480-49c0-be60-0084b31a0942",
    "status": "IN_PROGRESS",
    "message": "",
    "action": "APPLY",
    "stack_id": "ab100110-e014-11e7-85f3-c3bc8b9fdda1",
    "applied_status": "IN_PROGRESS",
    "applied_action": "APPLY",
    "data": {
      "stack_id": "ab100110-e014-11e7-85f3-c3bc8b9fdda1",
      "stack_job_id": "5a3148926e53230020466174",
      "stack_job_is_latest": true,
      "action": "APPLY",
      "status": "IN_PROGRESS",
      "tenantId": "a8fbbe67-1480-49c0-be60-0084b31a0942"
    }
  }
}

```

Destroy a template instance (stack)

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https:///${TSA_HOSTNAME}/cam/api/v1/stacks/${STACK_ID}/delete?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Example response:

```

{
  "name": "vSphere95",
  "created_at": "2017-11-06T20:18:11.697Z",
  "templateId": "5a00a2448af3f3002010aae1",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "prebuilt",
  "templateProvider": "VMware vSphere",
  "template_type": "Terraform",
  "template_format": "HCL",
  "applied_state_id": "5a00c4a08af3f3002010ab57",
  "latest_state_id": "5a00c4a08af3f3002010ab57",
  "cloud_connection_ids": [
    "5a00ab148af3f3002010ab45"
  ],
  "id": "5a00c3838af3f3002010ab53",
  "tenantId": "df87bf8a-2ea9-478d-9b37-2b36da4af6ed",
  "status": "SUCCESS",
  "message": "",
  "action": "DELETE",
  "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
  "applied_status": "SUCCESS",
  "applied_action": "DELETE",

```

```
"data": {
  "stack_id": "9cb55c20-c32f-11e7-8d7f-d1b85b52d79c",
  "stack_job_id": "5a00c4a04e13b5001f4aec2",
  "stack_job_is_latest": true,
  "status": "SUCCESS",
  "action": "DELETE",
  "message": "\nInitializing provider plugins...\nTerraform has been
successfully initialized!\n\nYou may now begin working with Terraform.
```

Delete a template instance (stack) after destroying it

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${STACK_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example response: {"count":1}

Retrieve stack information

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/{STACK_ID}/retrieve?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

This retrieves information about the stack including its parameters. However, if any of those parameters have **secured:true**, then the values for those parameters are hidden, that is, replaced with asterisks by default.

To return secured values back in plain text, run the POST with **showSecured=true** parameter.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/{STACK_ID}/retrieve?
showSecured=true&tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGu
id=${NAMESPACE}"
```

Download the Terraform state (tfstate) for a template instance

This API downloads the Terraform state (tfstate) of a template instance.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-o terraform.tfstate \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_STACK_ID}/tfstate?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Note: The **-o** file option is optional. It writes the output to a file instead of stdout.

Download the Terraform state (tfstate) for a specific stack state

This API downloads the Terraform state (tfstate) for a specific stack state.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-o terraform.tfstate \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_STACK_ID}/stackstates/${CAM_STACK_STATE_ID}/tfstate?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Upload the Terraform state (tfstate) for a template instance

This API uploads the Terraform state (tfstate) to a template instance.

You must have Administrator, cluster administrator, or editor role to run this API.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-F 'tfstate=@your_tfstate_file' \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_STACK_ID}/tfstate?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Download the execution plan for a template instance

This API downloads an execution plan for a template instance.

You must have Administrator, cluster administrator, or editor role to run this API.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-o terraform_execution_plan \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_STACK_ID}/executionPlan?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Note: The execution plan file is base64 encoded. You must decode the file content before using it.

Retrieve the execution plan in json format for a template instance

This API retrieves the execution plan in json format for a template instance. The API works only for template instances that use Terraform engine 0.12.x or newer.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_STACK_ID}/executionPlan/show?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Terraform version APIs

This section includes Terraform version APIs.

- [Retrieve Terraform versions](#)
- [Create a Terraform version](#)
- [Update a Terraform version](#)

- [Delete a Terraform version](#)
- [Set Terraform version as Default](#)
- [Enable/disable a Terraform version](#)

Retrieve Terraform versions

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions?
tenantId=${TSA_TENANT_ID}"
```

Create a Terraform version

You must have Administrator or cluster administrator role to run this API.

```
curl -k -X POST -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-F "template_type=Terraform" \
-F "version=0.15.5" \
-F "default=false" \
-F "enabled=true" \
-F "zipfile=@terraform_0.15.5.zip" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Update a Terraform version

You must have Administrator or cluster administrator role to run this API.

```
curl -k -X PUT \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @update_terraform_version.json \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions/${TERRAFORM_VERSION_ID}
?tenantId=${TSA_TENANT_ID}"
```

Example of the request body:

```
{
  version: '0.12.15',
  default: true
}
```

Delete a Terraform version

You must have Administrator or cluster administrator role to run this API.

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions/${TERRAFORM_VERSION_ID}
?tenantId=${TSA_TENANT_ID}"
```

Set Terraform version as default

You must have Administrator or cluster administrator role to run this API.

```
curl -k -X PUT \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @default_terraform_version.json \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions/${TERRAFORM_VERSION_ID}
?tenantId=${TSA_TENANT_ID}"
```

Example of the request body:

```
{
  default: true
}
```

Enable/disable a Terraform version

You must have Administrator or cluster administrator role to run this API.

```
curl -k -X PUT \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @enable_terraform_version.json \
"https://${TSA_HOSTNAME}/cam/api/v1/providerengineversions/${TERRAFORM_VERSION_ID}
?tenantId=${TSA_TENANT_ID}"
```

Example of the request body to enable the Terraform version:

```
{
  enabled: true
}
```

Example of the request body to disable the Terraform version:

```
{
  enabled: false
}
```

Terraform Provider Engine file APIs

The following section includes Terraform Provider Engine file APIs. You can upload Terraform plug-in files to Terraform pods by using these APIs.

- [Retrieve Terraform system and user plug-ins](#)
- [Retrieve user-uploaded Terraform plug-ins](#)
- [Create a Terraform Provider Engine file](#)
- [Delete a Terraform Provider Engine file](#)

Retrieve Terraform system and user plug-ins

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerenginefiles?tenantId=${TSA_TENANT_ID}"
```


Retrieve user-uploaded Terraform plug-ins

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerenginefiles?filter[where]
[category]=user&tenantId=${TSA_TENANT_ID}"
```

Create a Terraform Provider Engine file

You must have an Administrator or cluster administrator role to run this API.

```
curl -k -X POST -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-F "template_type=Terraform" \
-F "type=file" \
-F "zipfile=@customer_files.zip" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerenginefiles?
tenantId=${TSA_TENANT_ID}"
```

In the previous example, the user uploads a new Terraform plug-in. The archive can contain multiple plug-ins that are copied to the following path on each provider Terraform pod when they are restarted.

```
/home/terraform/.terraform.d/plugins
```

Delete a Terraform Provider Engine file

You must have an Administrator or cluster administrator role to run this API.

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/providerenginefiles/${PROVIDERENGINEFILES_ID}?
tenantId=${TSA_TENANT_ID}"
```

Resource-specific APIs

This section includes resource specific APIs with examples.

- [Restart OS](#)
- [Import VM by IDs](#)
- [Import VM by Name](#)
- [Create snapshots](#)
- [Revert to snapshot](#)
- [Delete a snapshot](#)
- [Revert to current snapshot](#)
- [Get snapshots](#)
- [Taint a resource from a stack](#)
- [Untaint a resource from a stack](#)
- [Retrieve and change power states for resources](#)
- [Start a virtual machine](#)
- [Stop a virtual machine](#)
- [Reset virtual machine](#)

- [Get the status of a stopping or starting a virtual machine](#)
- [Retrieve all IaaS resource settings](#)
- [Retrieve a specific IaaS resource settings](#)
- [Create IaaS resource settings](#)
- [Update IaaS resource settings](#)
- [Delete IaaS resource settings](#)

Note: There are three possible statuses for a resource task:

- IN_PROGRESS
- SUCCESS
- FAILED

Restart OS

The API call restarts the OS. It is available only for VMware vSphere virtual machine actions. The option is enabled only when the virtual machine powerstate is ON.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name": "restart"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

POST method.

Body: { "task_name": "restart" }

Example response:

```
{
  "cloud_resource_id": "421ac557-7793-fdd5-7b83-c3525571abfe",
  "task_name": "restart",
  "status": "IN_PROGRESS",
  "created": "2017-11-13T16:47:58.298Z",
  "updated": "2017-11-13T16:47:58.298Z",
  "stack_id": "7d55d130-c239-11e7-b95f-c5cc4d245984",
  "resource_id": "59ff272d4ba47e0020cb6583",
  "resource_name": "vsphere_virtual_machine.vm_1",
  "id": "5a09ccbe9fe5890020316467"
}
```

Import VM by IDs

Imports an existing virtual machine by using the ID. You can view stack instance in the template deployed instances page.

```
https://${TSA_HOSTNAME}/cam/api/v1/stacks/import?
tenantId=${TSA_TENANT_ID}&cloudOE_spaceGuid=${NAMESPACE}
```

POST method

For example, import a virtual machine's resources that are available in stack "fit-vm17-76":

Body

```
{
  "name": "fit-vm17-76",
  "cloudConnectionId": "5a8983938f22883ee775",
  "templateId": "5b39493e3984938943eb8b6",
  "templateVersionId": "5b39493e3984938943eb8c5",
  "idFromProvider": "/Rack-8283_1000167/vm/shared/fit-vm17-76"
}
```

Where,

- **name** is the name of the stack.
- **cloudConnectionId** is the ID of the connection where the virtual machine can be found. Use the credentials used in that cloud to find the virtual machine.
- **templateId** is to associate a template to a stack instance server.
- **templateVersionId** is the version of the template. It is an optional parameter.
- **idFromProvider** is the id of the virtual machine. The format differs for PowerVC and OpenStack.
- **resourceName** is an optional parameter. If resourceName is set, then you can import from providers other than vSphere and OpenStack. If you do not use this parameter, then the default values are as follows:
 - openstack_compute_instance_v2.vm_1 for OpenStack instances
 - vsphere_virtual_machine.vm_1 for VMWare vSphere instances

Import VM by Name

Imports an existing virtual machine by using the name. You can view the stack instance in the template deployed instances page.

```
https://${TSA_HOSTNAME}/cam/api/v1/stacks/import?
tenantId=${TSA_TENANT_ID}&cloudOE_spaceGuid=${NAMESPACE}
```

POST method

For example, import a virtual machine resource from stack "fit-vm17-76":

Body

```
{
  "name": "fit-vm17-76",
  "cloudConnectionName": "fit3",
  "templateName": "Import existing VM with a single disk - VMware vSphere",
  "templateVersionName": "1.0",
  "idFromProvider": "/Rack-8283_1000167/vm/Shared/fit-vm17-76"
}
```

Where,

- **name** is the name of the stack.
- **CloudConnectionName** is the connection name where the virtual machine can be found. Use the credentials used in that cloud to find the virtual machine.
- **templateName** is to associate a template to a stack instance server. It can be VMware vSphere or PowerVC/OpenStack.
- **templateVersionName** is the version of the template. It is an optional parameter.
- **idFromProvider** is the id of the virtual machine. The format differs for PowerVC and OpenStack.
- **resourceName** is an optional parameter. If resourceName is set, then you can import from providers other than vSphere and OpenStack. If you do not use this parameter, then the default values are as follows:
 - openstack_compute_instance_v2.vm_1 for OpenStack instances

- vsphere_virtual_machine.vm_1 for VMWare vSphere instances

Create snapshot

This API creates a virtual machine snapshot for a VMware instance.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

Sample URL:

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name": "snapshot", "parameters": [ \
  { \
    "name": "snapshot_name", \
    "value": ${SNAPSHOT_NAME} \
  }, \
  { \
    "name": "quiesce", \
    "value": true || false \
  }, \
  { \
    "name": "memory", \
    "value": true || false \
  }, \
  { \
    "name": "description", \
    "value": ${DESCRIPTION} # Optional \
  } \
]}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

POST method

BODY:

```
{
  "task_name": "snapshot",
  "parameters": [
    {
      "name": "snapshot_name",
      "value": "NAME"
    },
    {
      "name": "quiesce",
      "value": false
    },
    {
      "name": "memory",
      "value": false
    },
    {
      "name": "description",
      "value": "DESCRIPTION"
    }
  ]
}
```

Note:

- The long ID in the URL after **iaasResources** is the RESOURCE_ID or the ID of the virtual machine.
- The description is an optional parameter.

Revert to a snapshot

This API reverts to the selected snapshot.

Note:

- The virtual machine **resource.id** can be retrieved from the [update the state of the template instance API](#).
- The **snapshot_id** can be retrieved from the [Get snapshots API](#).

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name": "revertToSnapshot", "parameters": [ \
  { \
    "name": "snapshot_id", \
    "value": "${SNAPSHOT_ID} \
  } \
]}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

POST method

Sample BODY:

```
{
  "task_name": "revertToSnapshot",
  "parameters": [
    {
      "name": "snapshot_id",
      "value": "snapshot-19477"
    }
  ]
}
```

Note: The long ID in the URL after **iaasResources** is the RESOURCE_ID or the ID of the virtual machine.

Delete snapshot

This API deletes the selected snapshot.

Note:

- The virtual machine **resource.id** can be retrieved from the [update the state of the template instance API](#).
- The **snapshot_id** can be retrieved from the [Get snapshots API](#).

Sample URL:

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
```

```
-H "Content-Type: application/json" -d '{"task_name":"deleteSnapshot",
"parameters": [ \
  { \
    "name": "snapshot_id", \
    "value": "${SNAPSHOT_ID} \
  } \
]}' "https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

POST method

Sample BODY:

```
{
  "task_name": "deleteSnapshot",
  "parameters": [
    {
      "name": "snapshot_id",
      "value": "snapshot-19477"
    }
  ]
}
```

Note: The long ID in the URL after `iaasResources` is the `RESOURCE_ID` or the ID of the virtual machine.

Revert to current snapshot

This API restores to the parent snapshot of the current state of a virtual machine.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" -d '{"task_name":"revertToCurrent"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Sample URL:

`https://cam-proxy/cam/api/v1/iaasresources/5c47740ef76392001eb4d073/tasks`

POST method

BODY:

```
{
  "task_name": "revertToCurrent"
}
```

Note: The long ID in the URL after `iaasResources` is the `RESOURCE_ID` or the ID of the virtual machine.

Get snapshots

This API retrieves a list of snapshots.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/query/getsnapshots?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Sample URL:

```
https://cam-proxy/cam/api/v1/iaasresources/5c47740ef76392001eb4d073/query/getsnapshots
```

GET method

Note: The long ID in the URL after `iaasResources` is the `RESOURCE_ID` or the ID of the virtual machine.

Taint a resource from a stack

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
export CAM_RESOURCE_ID=<resource.id>
```

Note: The `resource.id` can be retrieved from [Update state of template instance](#) POST `/stacks/id/retrieve` call.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_TEMPLATE_ID}/taint/${CAM_RESOURCE_ID}?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example output:

```
{
  "name": "vSphere95",
  "created_at": "2018-01-10T18:35:52.140Z",
  "templateId": "5a4fe8f98bec4000200dc310",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "userCreated",
  "templateProvider": "IBM",
  "template_type": "Terraform",
  "applied_state_id": "5a565d4d8bec4000200dc32f",
  "latest_state_id": "5a56600a8bec4000200dc331",
  "cloud_connection_ids": [
    "5a3035908bec4000200dc30d"
  ],
  "id": "5a565d088bec4000200dc32c",
  "tenantId": "cd3ed377-8167-4078-b486-309b1f237928",
  "status": "IN_PROGRESS",
  "message": "",
  "action": "TAINT",
  "stack_id": "1616c4e0-f635-11e7-b2c8-2b0dbae0c060",
  "applied_status": "SUCCESS",
  "applied_action": "APPLY",
  "data": {
    "stack_id": "1616c4e0-f635-11e7-b2c8-2b0dbae0c060",
    "stack_job_id": "5a56600a035d8000204e8f27",
    "stack_job_is_latest": true,
    "action": "TAINT",
```

```

    "status": "IN_PROGRESS",
    "tenantId": "cd3ed377-8167-4078-b486-309b1f237928"
  }
}

```

Untaint a resource from a stack

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks/${CAM_TEMPLATE_ID}/untaint/${CAM_RESOURCE_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Example output:

```

{
  "name": "vSphere95",
  "created_at": "2018-01-10T18:35:52.140Z",
  "templateId": "5a4fe8f98bec4000200dc310",
  "namespaceId": "default",
  "templateName": "SingleVirtualMachine",
  "templateCategory": "userCreated",
  "templateProvider": "IBM",
  "template_type": "Terraform",
  "applied_state_id": "5a565d4d8bec4000200dc32f",
  "latest_state_id": "5a56618b8bec4000200dc332",
  "cloud_connection_ids": [
    "5a3035908bec4000200dc30d"
  ],
  "id": "5a565d088bec4000200dc32c",
  "tenantId": "cd3ed377-8167-4078-b486-309b1f237928",
  "status": "IN_PROGRESS",
  "message": "",
  "action": "UNTAINT",
  "stack_id": "1616c4e0-f635-11e7-b2c8-2b0dbae0c060",
  "applied_status": "SUCCESS",
  "applied_action": "APPLY",
  "data": {
    "stack_id": "1616c4e0-f635-11e7-b2c8-2b0dbae0c060",
    "stack_job_id": "5a56618b035d8000204e8f28",
    "stack_job_is_latest": true,
    "action": "UNTAINT",
    "status": "IN_PROGRESS",
    "tenantId": "cd3ed377-8167-4078-b486-309b1f237928"
  }
}

```

Retrieve and change power states for resources

This API retrieves and changes power states for resources.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \

```



```
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/query/getPowerState?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

GET method

```
Response: {
  "poweredOn": true || false
}
```

Start a virtual machine

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name": "start"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example response

```
{
  "cloud_resource_id": "421ac557-7793-fdd5-7b83-c3525571abfe",
  "task_name": "start",
  "status": "IN_PROGRESS",
  "created": "2017-11-13T16:51:37.972Z",
  "updated": "2017-11-13T16:51:37.972Z",
  "stack_id": "7d55d130-c239-11e7-b95f-c5cc4d245984",
  "resource_id": "59ff272d4ba47e0020cb6583",
  "resource_name": "vsphere_virtual_machine.vm_1",
  "id": "5a09cd999fe5890020316468"
}
```

Stop a virtual machine

```
export CAM_RESOURCE_ID=<resource.id>
```

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name": "stop"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example response

```
{
  "cloud_resource_id": "421ac557-7793-fdd5-7b83-c3525571abfe",
  "task_name": "stop",
  "status": "IN_PROGRESS",
  "created": "2017-11-13T16:47:58.298Z",
}
```

```

"updated": "2017-11-13T16:47:58.298Z",
"stack_id": "7d55d130-c239-11e7-b95f-c5cc4d245984",
"resource_id": "59ff272d4ba47e0020cb6583",
"resource_name": "vsphere_virtual_machine.vm_1",
"id": "5a09ccbe9fe5890020316467"
}

```

Reset a virtual machine

You can reset and restart a VMware virtual machine.

Note: The virtual machine `resource.id` can be retrieved from the [update the state of the template instance API](#).

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-d '{"task_name":"reset"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Example response

```

{
  "cloud_resource_id": "421ac557-7793-fdd5-7b83-c3525571abfe",
  "task_name": "reset",
  "status": "IN_PROGRESS",
  "created": "2017-11-13T16:47:58.298Z",
  "updated": "2017-11-13T16:47:58.298Z",
  "stack_id": "7d55d130-c239-11e7-b95f-c5cc4d245984",
  "resource_id": "59ff272d4ba47e0020cb6583",
  "resource_name": "vsphere_virtual_machine.vm_1",
  "id": "5a09ccbe9fe5890020316467"
}

```

Get the status of a stopping or starting a virtual machine

```
export CAM_IAASTASK_ID=<response.id>
```

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json" \
-d '{"task_name":"stop"}' \
"https://${TSA_HOSTNAME}/cam/api/v1/iaasresources/${CAM_RESOURCE_ID}/tasks/${CAM_I
AASTASK_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Example response:

```

{
  "cloud_resource_id": "421ac557-7793-fdd5-7b83-c3525571abfe",
  "task_name": "stop",
  "status": "SUCCESS",
  "created": "2017-11-13T16:47:58.298Z",
  "updated": "2017-11-13T16:48:03.728Z",
  "stack_id": "7d55d130-c239-11e7-b95f-c5cc4d245984",
  "resource_id": "59ff272d4ba47e0020cb6583",
  "resource_name": "vsphere_virtual_machine.vm_1",
  "message": "",

```

```
"id": "5a09ccbe9fe5890020316467"
}
```

Retrieve all IaaS resource settings

This API allows you to get all Terraform resource settings.

```
curl -k -X GET \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/api/v1/iaasresourcesettings?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Note: The cloudOE_spaceGuid query parameter is optional and the search results display all IaaS resource settings that \${NAMESPACE} resolves to.

Retrieve a specific IaaS resource settings

This API allows you to show the details about a Terraform resource setting.

```
curl -k -X GET \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \

"https://${TSA_HOSTNAME}/cam/api/v1/iaasresourcesettings/${IAAS_RESOURCE_SETTINGS_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Create IaaS resource settings

This API allows an administrator to hide specific information about the Terraform resource.

```
curl -k -X POST \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  -H "Content-Type: application/json" \
  "https://${TSA_HOSTNAME}/cam/api/v1/iaasresourcesettings?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}" \
  -d '{
    "type": "aws_virtual_machine",
    "attributes": ["cpu_reservation", "guest_ip_addresses"],
    "hidden": true
  }'
```

This example hides two attributes from all the **aws_virtual_machine** resources. In case you want to hide all the attributes from all the resource types, create an IaaS resource settings with type as **__all__**.

Request body parameters:

- **type** - the provider resource type (that is, **vpshere_virtual_machine**). This is not the resource name.
- **attributes** - list of attributes to be hidden. This parameter is optional. In case it is omitted, all the attributes for that specific resource type are hidden.
- **hidden** - It accepts true or false depending on whether you want to hide or show the attribute.

Note: The cloudOE_spaceGuid query parameter is optional and the IaaS resource setting is created in the namespace that \${NAMESPACE} resolves to.

Update IaaS resource settings

This API allows the cluster administrator and administrator to modify the settings for a Terraform resource type.

```
curl -k -X PUT \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  -H "Content-Type: application/json" \
  -d '{"type": "aws_virtual_machine"}' \

"https://${TSA_HOSTNAME}/cam/api/v1/iaasresourcesettings/${IAAS_RESOURCE_SETTINGS_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Delete IaaS resource settings

This API allows the cluster administrator and administrator to delete settings for a Terraform resource type.

```
curl -k -X DELETE \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \

"https://${TSA_HOSTNAME}/cam/api/v1/iaasresourcesettings/${IAAS_RESOURCE_SETTINGS_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Service and service instance APIs

This section includes all service and service instance APIs with examples.

- [Service APIs](#)
 - [Create a service with default version](#)
 - [Add a service version to an existing service](#)
 - [Set a default version for a service](#)
 - [Edit a Service version](#)
 - [Delete a Service version in a service](#)
 - [Delete a Service \(with all its versions\)](#)
 - [GET the list of services](#)
 - [Get Service versions for a service using `service-id`](#)
 - [Get Service version manifest by service version ID](#)
 - [Publish a service version](#)
 - [Duplicate a service](#)
 - [Change the category of a service](#)
 - [Rename a service](#)
 - [Change service image for a service](#)
 - [Assign access to a service](#)
 - [Retire a service version](#)
 - [Unretire a service version](#)
 - [Auto-retire a service version](#)
- [Service Instance APIs](#)
 - [Deploy a Service Instance](#)
 - [Get the list of deployed instance: GET call](#)
 - [Get details of a Service Instance by using the id](#)
 - [Get input parameters of a Service Instance](#)
 - [Edit Service Instance](#)
 - [Retry Service Instance](#)

- [Resume Service Instance](#)
 - [Upgrade or downgrade a Service Instance](#)
 - [Bind a service instance](#)
 - [Unbind Instance](#)
 - [Get custom actions](#)
 - [Invoke custom action](#)
 - [Terminate Service Instance](#)
 - [Cleanup Service Instance](#)
 - [Cleanup Service Instance in inprogress state](#)
- [Standalone action APIs](#)
 - [Create Standalone Action](#)
 - [Invoke Standalone Action](#)
- [ICON APIs](#)
 - [Upload service icons](#)
 - [Delete service icons](#)
 - [Retrieve all service icons](#)
 - [Download service icon](#)

Service APIs

Create a service with default version

This API call creates a service version and adds a default version. The service and the service version are returned in response.

```
curl -k -X POST \
  'https://{TSA_HOSTNAME}/cam/composer/api/v1/Service/createService?
  ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @service_create.json
```

Payload, content of 'service_create.json'

```
{
  "service": {
    "specVersion": "v3",
    "catalog_metadata": {
      "name": "SampleV1",
      "description": "A Sample Single Virtual Machine Service",
      "image": "serviceicon_1.svg",
      "category": "",
      "bullets": [
      ],
      "providerDisplayName": "IBM",
      "longDescription": "",
      "documentationUrl": "",
      "supportUrl": "",
      "bindable": false,
      "updatable": "",
      "systemTags": true,
      "author": "admin"
    },
    "tags": [
      {
        "name": "request_group",
```

```

        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user at
time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"actions": [
    {
        "type": "provision",

```

```

"name": "Provision",
"description": "Default action for deployment of service",
"input_parameters": [
  {
    "name": "vmMemory",
    "label": "vmMemory",
    "customtype": "string",
    "type": "string",
    "immutable": false,
    "hidden": false,
    "required": false,
    "secured": false
  },
  {
    "name": "vmCpu",
    "label": "vmCpu",
    "customtype": "string",
    "type": "string",
    "immutable": false,
    "hidden": false,
    "required": false,
    "secured": false
  }
],
"flow": {
  "conditions": [

```

```

],
"templates": [
  {
    "SingleVirtualMachine": {
      "title": "SingleVirtualMachine_cca453",
      "template_name": "SingleVirtualMachine",
      "version": "2.3",
      "id": "singlevi23cd3149",
      "template_type": "Terraform",
      "template_content_type": "VMware vSphere",
      "template_provider": "VMware vSphere",
      "instance_name": "",
      "cloud_connection_name": "vmware",
      "template_data_objects": {

```

```

      },
      "template_params": {
        "vm_1_name": "ver2vm",
        "vm_1_dns_servers": [
          "1.1.1.2"
        ],
        "vm_1_dns_suffixes": [
          "test.cam.com"
        ],
        "vm_1_root_disk_keep_on_remove": "false",
        "vm_1_domain": "test.cam.com",
        "vm_1_image": "terraform/template/rhel_7_2_small",
        "vm_1_root_disk_size": "25",
        "vm_1_root_disk_controller_type": "scsi",
        "vm_1_root_disk_datastore": "NSXcluster/datastore1",
        "vm_1_root_disk_type": "eager_zeroed",
        "vm_1_ipv4_gateway": "10.4.10.1",
        "vm_1_ipv4_address": "10.4.10.17",
        "vm_1_memory": "${input_parameters.vmMemory}",
        "vm_1_ipv4_prefix_length": "23",

```

```

        "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
        "vm_1_network_interface_label": "external-net-410",
        "vm_1_cluster": "icm-vmware-nsx-2",
        "vm_1_datacenter": "NSX Manage To",
        "vm_1_folder": "terraform/test",
        "vm_1_adapter_type": "vmxnet3",
        "vm_1_resource_pool": "cam"
    },
    "outputs": [
    ],
    "immutable": false
}
}
],
"resthooks": [
],
"notifications": [
],
"sequence": {
    "0": "singlevi23cd3149"
},
"error_sequence": {
}
},
"output_parameters": [
]
}
],
"plans": [
{
    "name": "Standard",
    "description": "To deploy a Standard plan",
    "actions": [
    ],
    "plan_parameters": [
    ]
}
]
}
}
}

```

Response

```

{
  "id": "5df37c4aa051bd001d161161",
  "name": "SampleV1",
  "author": "admin",
  "image": "serviceicon_1.svg",
  "status": "",
  "serviceManifestId": "5df37c4ba051bd001d161162",
  "tag": "5c6e73077077fc001715703b",
  "creationTime": "2019-12-13T11:55:51.389Z",
  "lastUpdated": "2019-12-13T11:55:55.078Z",
  "namespaceId": "default",
  "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
  "description": "A Sample Single Virtual Machine Service",
  "longDescription": "",
  "defaultServiceVersion": {
    "id": "5df37c4ba051bd001d161162",
    "version": "1.0.0.0",
    "ServiceEngine_InputFile": {
      "service": {
        "specVersion": "v3",
        "catalog_metadata": {

```



```

    "name": "SampleV1",
    "description": "A Sample Single Virtual Machine Service",
    "image": "serviceicon_1.svg",
    "category": "CloudServices",
    "bullets": [],
    "providerDisplayName": "IBM",
    "longDescription": "",
    "documentationUrl": "",
    "supportUrl": "",
    "bindable": false,
    "updatable": "",
    "systemTags": true,
    "author": "admin"
  },

  "tags": [
    {
      "name": "request_group",
      "label": "request_group",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
      "isSystemTag": true,
      "default": "${svc_instance.group}",
      "customtype": "string",
      "permission": "Read-Only"
    },

    {
      "name": "request_user",
      "label": "request_user",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The user id of the current user that requested the provisioning of the template.",
      "isSystemTag": true,
      "default": "${svc_instance.owner}",
      "customtype": "string",
      "permission": "Read-Only"
    },

    {
      "name": "service_name",
      "label": "service_name",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Name provided for the service instance by the end user at time of request",
      "isSystemTag": true,
      "default": "${svc_instance.name}",
      "customtype": "string",
      "permission": "Read-Only"
    }
  ]
}

```

```

    },
    {
      "name": "service_identifier",
      "label": "service_identifier",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
      "isSystemTag": true,
      "default": "${svc_instance.service_instance_id}",
      "customtype": "string",
      "permission": "Read-Only"
    }
  ],
  "actions": [
    {
      "type": "provision",
      "name": "Provision",
      "description": "Default action for deployment of service",
      "input_parameters": [
        {
          "name": "vmMemory",
          "label": "vmMemory",
          "customtype": "string",
          "type": "string",
          "immutable": false,
          "hidden": false,
          "required": false,
          "secured": false
        },
        {
          "name": "vmCpu",
          "label": "vmCpu",
          "customtype": "string",
          "type": "string",
          "immutable": false,
          "hidden": false,
          "required": false,
          "secured": false
        }
      ]
    },
    {
      "flow": {
        "conditions": [],
        "templates": [
          {
            "SingleVirtualMachine": {
              "title": "SingleVirtualMachine_cc453",
              "template_name": "SingleVirtualMachine",
              "version": "2.3",
              "id": "singlevi23cd3149",
              "template_type": "Terraform",
              "template_content_type": "VMware vSphere",
              "template_provider": "VMware vSphere",
              "instance_name": "",
              "cloud_connection_name": "vmware",
              "template_data_objects": {},
              "template_params": {
                "vm_1_name": "ver2vm",
                "vm_1_dns_servers": [

```

```

        "1.1.1.2"
    ],
    "vm_1_dns_suffixes": [
        "test.cam.com"
    ],
    "vm_1_root_disk_keep_on_remove": "false",
    "vm_1_domain": "test.cam.com",
    "vm_1_image": "terraform/template/rhel_7_2_small",
    "vm_1_root_disk_size": "25",
    "vm_1_root_disk_controller_type": "scsi",
    "vm_1_root_disk_datastore": "NSXcluster/datastore1",
    "vm_1_root_disk_type": "eager_zeroed",
    "vm_1_ipv4_gateway": "10.4.10.1",
    "vm_1_ipv4_address": "10.4.10.17",
    "vm_1_memory": "${input_parameters.vmMemory}",
    "vm_1_ipv4_prefix_length": "23",
    "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
    "vm_1_network_interface_label": "external-net-410",
    "vm_1_cluster": "icm-vmware-nsx-2",
    "vm_1_datacenter": "NSX Manage To",
    "vm_1_folder": "terraform/test",
    "vm_1_adapter_type": "vmxnet3",
    "vm_1_resource_pool": "cam"
},

    "outputs": [],
    "immutable": false
}
],

    "resthooks": [],
    "notifications": [],
    "sequence": {
        "0": "singlevi23cd3149"
    },
    "error_sequence": {}
},
    "output_parameters": []
}
],
"plans": [
    {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
    }
],
"version": "1.0.0.0"
}
},
"service_id": "5df37c4aa051bd001d161161",
"namespaceId": "default",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,

```

```

        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user at
time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"systemTags": true,
"creationTime": "2019-12-13T11:55:54.970Z"
},
"warning": [
    "No 'Category' is provided in the source code. Assigning the service to
'CloudServices' \n",
    "No 'version' is provided in the source code. Assigning the service, the

```

```
default version '1.0.0.0' \n"
  ]
}
```

Add a service version to an existing service

Service name, category, and image cannot be changed by using this API.

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/addServiceVersion?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_version_add.json
```

Payload: Content of `service_version_add.json`

```
{
  "service": {
    "version": "1.0.0.1",
    "specVersion": "v3",
    "catalog_metadata": {
      "name": "SampleV1",
      "description": "A Sample Single Virtual Machine Service",
      "image": "serviceicon_1.svg",
      "category": "CloudServices",
      "bullets": [],
      "providerDisplayName": "IBM",
      "longDescription": "",
      "documentationUrl": "",
      "supportUrl": "",
      "bindable": false,
      "updatable": "",
      "systemTags": true,
      "author": "admin"
    },
  },
  "tags": [
    {
      "name": "request_group",
      "label": "request_group",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
      "isSystemTag": true,
      "default": "${svc_instance.group}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "request_user",
      "label": "request_user",
      "type": "string",
      "immutable": true,
      "hidden": false,
```

```

    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "default": "${svc_instance.owner}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user at time
of request",
    "isSystemTag": true,
    "default": "${svc_instance.name}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
    "isSystemTag": true,
    "default": "${svc_instance.service_instance_id}",
    "customtype": "string",
    "permission": "Read-Only"
  }
],
"actions": [
  {
    "type": "provision",
    "name": "Provision",
    "description": "Default action for deployment of service",
    "input_parameters": [
      {
        "name": "vmMemory",
        "label": "vmMemory",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false
      },
      {
        "name": "vmCpu",
        "label": "vmCpu",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,

```

```

    "secured": false
  }
],
"flow": {
  "conditions": [],
  "templates": [
    {
      "SingleVirtualMachine": {
        "title": "SingleVirtualMachine_cca453",
        "template_name": "SingleVirtualMachine",
        "version": "2.3",
        "id": "singlevi23cd3149",
        "template_type": "Terraform",
        "template_content_type": "VMware vSphere",
        "template_provider": "VMware vSphere",
        "instance_name": "",
        "cloud_connection_name": "vmware",
        "template_data_objects": {},
        "template_params": {
          "vm_1_name": "ver2vm",
          "vm_1_dns_servers": [
            "1.1.1.2"
          ],
          "vm_1_dns_suffixes": [
            "test.cam.com"
          ],
          "vm_1_root_disk_keep_on_remove": "false",
          "vm_1_domain": "test.cam.com",
          "vm_1_image": "terraform/template/rhel_7_2_small",
          "vm_1_root_disk_size": "25",
          "vm_1_root_disk_controller_type": "scsi",
          "vm_1_root_disk_datastore": "NSXcluster/datastore1",
          "vm_1_root_disk_type": "eager_zeroed",
          "vm_1_ipv4_gateway": "10.4.10.1",
          "vm_1_ipv4_address": "10.4.10.17",
          "vm_1_memory": "${input_parameters.vmMemory}",
          "vm_1_ipv4_prefix_length": "23",
          "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
          "vm_1_network_interface_label": "external-net-410",
          "vm_1_cluster": "icm-vmware-nsx-2",
          "vm_1_datacenter": "NSX Manage To",
          "vm_1_folder": "terraform/test",
          "vm_1_adapter_type": "vmxnet3",
          "vm_1_resource_pool": "cam"
        },
      },
      "outputs": [],
      "immutable": false
    }
  ],
  "resthooks": [],
  "notifications": [
    {
      "emailnotification": {
        "title": "emailnotification_4278ce",
        "type": "email",
        "receiverList": "testmail@testr-mail.net",
        "id": "emailnot21410720",
        "sender": "CAMadmin",
        "copyToList": "",
        "subject": "VM Creation Succesful",
        "body": "VM create at ${templates.singlevi23cd3149.created_at}",
        "immutable": false
      }
    }
  ]
}

```

```

    }
  ],
  "sequence": {
    "0": "singlevi23cd3149",
    "1": "emailnot21410720"
  },
  "error_sequence": {},
},
"output_parameters": []
}
],
"plans": [
{
  "name": "Standard",
  "description": "To deploy a Standard plan",
  "actions": [],
  "plan_parameters": []
}
]
}
}

```

Response

```

{
  "id": "5df37f91a051bd001d161171",
  "version": "1.0.0.1",
  "ServiceEngine_InputFile": {
    "service": {
      "version": "1.0.0.1",
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "SampleV1",
        "description": "A Sample Single Virtual Machine Service",
        "image": "serviceicon_1.svg",
        "category": "CloudServices",
        "bullets": [],
        "providerDisplayName": "IBM",
        "longDescription": "",
        "documentationUrl": "",
        "supportUrl": "",
        "bindable": false,
        "updatable": "",
        "systemTags": true,
        "author": "admin"
      },
    },
    "tags": [
      {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
      },
    ],
  },

```



```

    {
      "name": "request_user",
      "label": "request_user",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The user id of the current user that requested the provisioning of the template.",
      "isSystemTag": true,
      "default": "${svc_instance.owner}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_name",
      "label": "service_name",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Name provided for the service instance by the end user at time of request",
      "isSystemTag": true,
      "default": "${svc_instance.name}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_identifier",
      "label": "service_identifier",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Unique identifier generated by ICAM which is mapped to the service instance of the template provisioned.",
      "isSystemTag": true,
      "default": "${svc_instance.service_instance_id}",
      "customtype": "string",
      "permission": "Read-Only"
    }
  ],
  "actions": [
    {
      "type": "provision",
      "name": "Provision",
      "description": "Default action for deployment of service",
      "input_parameters": [
        {
          "name": "vmMemory",
          "label": "vmMemory",
          "customtype": "string",
          "type": "string",
          "immutable": false,
          "hidden": false,
          "required": false,
          "secured": false
        },
        {
          "name": "vmCpu",

```

```

        "label": "vmCpu",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false
    }
],
"flow": {
    "conditions": [],
    "templates": [
        {
            "SingleVirtualMachine": {
                "title": "SingleVirtualMachine_cca453",
                "template_name": "SingleVirtualMachine",
                "version": "2.3",
                "id": "singlevi23cd3149",
                "template_type": "Terraform",
                "template_content_type": "VMware vSphere",
                "template_provider": "VMware vSphere",
                "instance_name": "",
                "cloud_connection_name": "vmware",
                "template_data_objects": {},
                "template_params": {
                    "vm_1_name": "ver2vm",
                    "vm_1_dns_servers": [
                        "1.1.1.2"
                    ],
                    "vm_1_dns_suffixes": [
                        "test.cam.com"
                    ],
                    "vm_1_root_disk_keep_on_remove": "false",
                    "vm_1_domain": "test.cam.com",
                    "vm_1_image": "terraform/template/rhel_7_2_small",
                    "vm_1_root_disk_size": "25",
                    "vm_1_root_disk_controller_type": "scsi",
                    "vm_1_root_disk_datastore": "NSXcluster/datastore1",
                    "vm_1_root_disk_type": "eager_zeroed",
                    "vm_1_ipv4_gateway": "10.4.10.1",
                    "vm_1_ipv4_address": "10.4.10.17",
                    "vm_1_memory": "${input_parameters.vmMemory}",
                    "vm_1_ipv4_prefix_length": "23",
                    "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
                    "vm_1_network_interface_label": "external-net-410",
                    "vm_1_cluster": "icm-vmware-nsx-2",
                    "vm_1_datacenter": "NSX Manage To",
                    "vm_1_folder": "terraform/test",
                    "vm_1_adapter_type": "vmxnet3",
                    "vm_1_resource_pool": "cam"
                },
                "outputs": [],
                "immutable": false
            }
        }
    ],
    "resthooks": [],
    "notifications": [
        {
            "emailnotification": {
                "title": "emailnotification_4278ce",
                "type": "email",
                "receiverList": "testmail@testr-mail.net",
                "id": "emailnot21410720",

```

```

        "sender": "CAMadmin",
        "copyToList": "",
        "subject": "VM Creation Succesful",
        "body": "VM create at ${templates.singlevi23cd3149.created_at}",
        "immutable": false
    }
}
],
"sequence": {
    "0": "singlevi23cd3149",
    "1": "emailnot21410720"
},
"error_sequence": {}
},
"output_parameters": []
}
],
"plans": [
{
    "name": "Standard",
    "description": "To deploy a Standard plan",
    "actions": [],
    "plan_parameters": []
}
]
}
},
"service_id": "5df37f85a051bd001d16116d",
"tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
"namespaceId": "default",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
{
    "name": "request_group",
    "label": "request_group",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
    "isSystemTag": true,
    "default": "${svc_instance.group}",
    "customtype": "string",
    "permission": "Read-Only"
},
{
    "name": "request_user",
    "label": "request_user",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "default": "${svc_instance.owner}",
    "customtype": "string",

```

```

    "permission": "Read-Only"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user at
time of request",
    "isSystemTag": true,
    "default": "${svc_instance.name}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
    "isSystemTag": true,
    "default": "${svc_instance.service_instance_id}",
    "customtype": "string",
    "permission": "Read-Only"
  }
],
"systemTags": true,
"creationTime": "2019-12-13T12:09:53.433Z",
"warning": []
}

```

Set a default version for a service

```

curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/updateServiceDe
tails?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
\
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @service_default_version.json

```

Payload sample1:

```
"default_version_id": "<<version id>>"
```

Payload sample2:

```

{
  "default_version_name": "1.0.0.1"
}

```

Response:

```
{
  "service": {
    "id": "5df38f28a051bd001d16117d",
    "name": "SampleV1",
    "serviceManifestId": "5df38f30a051bd001d161181",
    "author": "admin",
    "image": "serviceicon_1.svg",
    "tag": "5c6e73077077fc001715703b",
    "creationTime": "2019-12-13T13:16:23.237Z",
    "lastUpdated": "2019-12-13T13:25:01.646Z",
    "namespaceId": "default",
    "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
    "description": "A Sample Single Virtual Machine Service",
    "longDescription": ""
  }
}
```

Edit a Service Version

This API edits a service version. This API cannot change service name, category, and image.

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_VERSION_ID}/updateServiceVersion?ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_version_edit.json
```

Payload: content of service_version_edit.json

```
{
  "service": {
    "version": "1.0.0.1",
    "specVersion": "v3",
    "catalog_metadata": {
      "name": "SampleV1",
      "description": "A Sample Single Virtual Machine Service",
      "image": "serviceicon_1.svg",
      "category": "CloudServices",
      "bullets": [],
      "providerDisplayName": "IBM",
      "longDescription": "",
      "documentationUrl": "",
      "supportUrl": "",
      "bindable": false,
      "updatable": "",
      "systemTags": true,
      "author": "admin"
    },
    "tags": [
      {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may
```

```

be part of more than one group and this identifies the context of which group the
user made the request.",
  "isSystemTag": true,
  "default": "${svc_instance.group}",
  "customtype": "string",
  "permission": "Read-Only"
},
{
  "name": "request_user",
  "label": "request_user",
  "type": "string",
  "immutable": true,
  "hidden": false,
  "required": false,
  "secured": false,
  "description": "The user id of the current user that requested the
provisioning of the template.",
  "isSystemTag": true,
  "default": "${svc_instance.owner}",
  "customtype": "string",
  "permission": "Read-Only"
},
{
  "name": "service_name",
  "label": "service_name",
  "type": "string",
  "immutable": true,
  "hidden": false,
  "required": false,
  "secured": false,
  "description": "Name provided for the service instance by the end user at time
of request",
  "isSystemTag": true,
  "default": "${svc_instance.name}",
  "customtype": "string",
  "permission": "Read-Only"
},
{
  "name": "service_identifier",
  "label": "service_identifier",
  "type": "string",
  "immutable": true,
  "hidden": false,
  "required": false,
  "secured": false,
  "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
  "isSystemTag": true,
  "default": "${svc_instance.service_instance_id}",
  "customtype": "string",
  "permission": "Read-Only"
}
],
"actions": [
{
  "type": "provision",
  "name": "Provision",
  "description": "Default action for deployment of service",
  "input_parameters": [
    {
      "name": "vmMemory",
      "label": "vmMemory",
      "customtype": "string",
      "type": "string",

```

```

    "immutable": false,
    "hidden": false,
    "required": false,
    "secured": false
  },
  {
    "name": "vmCpu",
    "label": "vmCpu",
    "customtype": "string",
    "type": "string",
    "immutable": false,
    "hidden": false,
    "required": false,
    "secured": false
  }
],
"flow": {
  "conditions": [],
  "templates": [
    {
      "SingleVirtualMachine": {
        "title": "SingleVirtualMachine_cca453",
        "template_name": "SingleVirtualMachine",
        "version": "2.3",
        "id": "singlelevi23cd3149",
        "template_type": "Terraform",
        "template_content_type": "VMware vSphere",
        "template_provider": "VMware vSphere",
        "instance_name": "",
        "cloud_connection_name": "vmware",
        "template_data_objects": {},
        "template_params": {
          "vm_1_name": "ver2vm",
          "vm_1_dns_servers": [
            "1.1.1.2"
          ],
          "vm_1_dns_suffixes": [
            "test.cam.com"
          ],
          "vm_1_root_disk_keep_on_remove": "false",
          "vm_1_domain": "test.cam.com",
          "vm_1_image": "terraform/template/rhel_7_2_small",
          "vm_1_root_disk_size": "25",
          "vm_1_root_disk_controller_type": "scsi",
          "vm_1_root_disk_datastore": "NSXcluster/datastore1",
          "vm_1_root_disk_type": "eager_zeroed",
          "vm_1_ipv4_gateway": "10.4.10.1",
          "vm_1_ipv4_address": "10.4.10.17",
          "vm_1_memory": "${input_parameters.vmMemory}",
          "vm_1_ipv4_prefix_length": "23",
          "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
          "vm_1_network_interface_label": "external-net-410",
          "vm_1_cluster": "icm-vmware-nsx-2",
          "vm_1_datacenter": "NSX Manage To",
          "vm_1_folder": "terraform/test",
          "vm_1_adapter_type": "vmxnet3",
          "vm_1_resource_pool": "cam"
        },
        "outputs": [],
        "immutable": false,
        "provider_engine_version": "0.11.11"
      }
    }
  ]
},

```

```

"resthooks": [],
"notifications": [
  {
    "emailnotification": {
      "title": "emailnotification_4278ce",
      "type": "email",
      "receiverList": "testmail@test-mail.sample.net",
      "id": "emailnot21410720",
      "sender": "CAMadmin",
      "copyToList": "",
      "subject": "VM Creation Succesful",
      "body": "VM create at ${templates.singlevi23cd3149.created_at}",
      "immutable": false
    }
  },
  {
    "emailnotification": {
      "title": "emailnotification_ac336e",
      "type": "email",
      "receiverList": "testmail@test-mail.sample.net",
      "id": "emailnotd384eeec",
      "sender": "CAMadmin",
      "copyToList": "",
      "subject": "VM Creation failed",
      "body": "VM creation failure caused by : ${svc_instance.error.message}",
      "immutable": false
    }
  }
],
"sequence": {
  "0": "singlevi23cd3149",
  "1": "emailnot21410720"
},
"error_sequence": {
  "0": "emailnotd384eeec"
}
},
"output_parameters": []
}
],
"plans": [
  {
    "name": "Standard",
    "description": "To deploy a Standard plan",
    "actions": [],
    "plan_parameters": []
  }
]
}
}

```

Response:

```

{
  "id": "5df38f30a051bd001d161181",
  "version": "1.0.0.1",
  "ServiceEngine_InputFile": {
    "service": {
      "version": "1.0.0.1",
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "SampleV1",
        "description": "A Sample Single Virtual Machine Service",
        "image": "serviceicon_1.svg",

```



```

    "category": "CloudServices",
    "bullets": [],
    "providerDisplayName": "IBM",
    "longDescription": "",
    "documentationUrl": "",
    "supportUrl": "",
    "bindable": false,
    "updatable": "",
    "systemTags": true,
    "author": "admin"
  },
  "tags": [
    {
      "name": "request_group",
      "label": "request_group",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
      "isSystemTag": true,
      "default": "${svc_instance.group}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "request_user",
      "label": "request_user",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The user id of the current user that requested the provisioning of the template.",
      "isSystemTag": true,
      "default": "${svc_instance.owner}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_name",
      "label": "service_name",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Name provided for the service instance by the end user at time of request",
      "isSystemTag": true,
      "default": "${svc_instance.name}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_identifier",
      "label": "service_identifier",
      "type": "string",
      "immutable": true,

```

```

        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [
            {
                "name": "vmMemory",
                "label": "vmMemory",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            },
            {
                "name": "vmCpu",
                "label": "vmCpu",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            }
        ],
        "flow": {
            "conditions": [],
            "templates": [
                {
                    "SingleVirtualMachine": {
                        "title": "SingleVirtualMachine_cca453",
                        "template_name": "SingleVirtualMachine",
                        "version": "2.3",
                        "id": "singlevi23cd3149",
                        "template_type": "Terraform",
                        "template_content_type": "VMware vSphere",
                        "template_provider": "VMware vSphere",
                        "instance_name": "",
                        "cloud_connection_name": "vmware",
                        "template_data_objects": {},
                        "template_params": {
                            "vm_1_name": "ver2vm",
                            "vm_1_dns_servers": [
                                "1.1.1.2"
                            ],
                            "vm_1_dns_suffixes": [
                                "test.cam.com"
                            ],
                            "vm_1_root_disk_keep_on_remove": "false",
                            "vm_1_domain": "test.cam.com",
                            "vm_1-image": "terraform/template/rhel_7_2_small",

```

```

        "vm_1_root_disk_size": "25",
        "vm_1_root_disk_controller_type": "scsi",
        "vm_1_root_disk_datastore": "NSXcluster/datastore1",
        "vm_1_root_disk_type": "eager_zeroed",
        "vm_1_ipv4_gateway": "10.4.10.1",
        "vm_1_ipv4_address": "10.4.10.17",
        "vm_1_memory": "${input_parameters.vmMemory}",
        "vm_1_ipv4_prefix_length": "23",
        "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
        "vm_1_network_interface_label": "external-net-410",
        "vm_1_cluster": "icm-vmware-nsx-2",
        "vm_1_datacenter": "NSX Manage To",
        "vm_1_folder": "terraform/test",
        "vm_1_adapter_type": "vmxnet3",
        "vm_1_resource_pool": "cam"
    },
    "outputs": [],
    "immutable": false,
    "provider_engine_version": "0.11.11"
}
}
],
"resthooks": [],
"notifications": [
    {
        "emailnotification": {
            "title": "emailnotification_4278ce",
            "type": "email",
            "receiverList": "testmail@test-mail.sample.net",
            "id": "emailnot21410720",
            "sender": "CAMadmin",
            "copyToList": "",
            "subject": "VM Creation Succesful",
            "body": "VM create at ${templates.singlevi23cd3149.created_at}",
            "immutable": false
        }
    },
    {
        "emailnotification": {
            "title": "emailnotification_ac336e",
            "type": "email",
            "receiverList": "testmail@test-mail.sample.net",
            "id": "emailnotd384eeec",
            "sender": "CAMadmin",
            "copyToList": "",
            "subject": "VM Creation failed",
            "body": "VM creation failure caused by :
${svc_instance.error.message}",
            "immutable": false
        }
    }
],
"sequence": {
    "0": "singlevi23cd3149",
    "1": "emailnot21410720"
},
"error_sequence": {
    "0": "emailnotd384eeec"
}
},
"output_parameters": []
}
],
"plans": [

```

```

        {
            "name": "Standard",
            "description": "To deploy a Standard plan",
            "actions": [],
            "plan_parameters": []
        }
    ]
}
},
"service_id": "5df38f28a051bd001d16117d",
"tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
"namespaceId": "default",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user at
time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {

```

```

    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
    "isSystemTag": true,
    "default": "${svc_instance.service_instance_id}",
    "customtype": "string",
    "permission": "Read-Only"
  }
],
"systemTags": true,
"creationTime": "2019-12-13T13:16:32.250Z",
"serviceInstanceCount": 0,
"warning": []
}

```

Delete a Service version in a service

```

curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/deleteService?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
\
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_version_delete.json

```

Payload sample1:

```

{
  "version_id": "${VERSION_ID}"
}

```

Payload sample2:

```

{
  "version": "${VERSION}"
}

```

Delete a Service (with all its versions)

```

curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/deleteService?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
\
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_delete.json

```

Payload

```

{}

```

Response

```

{
  "count": "1"
}

```

```
}
```

GET the list of services

```
curl -k -X GET \  
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \  
"https://${TSA_HOSTNAME}/cam/composer/api/v1/service? \  
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

The service records that belong to the namespace value provided for `cloudOE_spaceGuid` are fetched as a result of this curl command. If you specify any value for `cloudOE_spaceGuid`, then all default services are fetched along with services that belong to globally accessible namespace.

Example response:

```
[{  
  "name": "helmmultiple",  
  "author": "admin",  
  "image": "serviceicon_1.svg",  
  "specVersion": "v2",  
  "status": "Draft",  
  "plans": {  
    "Standard": "5ab0dcb27bf798001e85724b"  
  },  
  "tags": [],  
  "parameters": [],  
  "serviceManifestId": "5ab0dcb27bf798001e857249",  
  "lastUpdated": "2018-03-20T10:04:34.051Z",  
  "published": "",  
  "creationTime": "Tue Mar 20 2018 10:04:34 GMT+0000 (UTC)",  
  "conditions": [{  
    "ibm-charts/ibm-nodejs-sample": {  
      "title": "ibm-charts/ibm-nodejs-sample",  
      "template_name": "ibm-nodejs-sample",  
      "version": "1.2.0",  
      "id": "ibmchartc99c9c4a",  
      "template_type": "Helm",  
      "template_content_type": "Cloud Pak for Multicloud Management",  
      "template_provider": "Cloud Pak for Multicloud Management",  
      "instance_name": "",  
      "cloud_connection_name": "MCM Connection",  
      "template_dataobjects": {},  
      "template_params": {  
        "namespace": "default",  
        "image-repository": "ibmcom/icp-nodejs-sample",  
        "image-tag": "latest",  
        "image-pullPolicy": "Always",  
        "livenessProbe-initialDelaySeconds": 30,  
        "livenessProbe-periodSeconds": 10,  
        "service-name": "Node",  
        "service-type": "NodePort",  
        "service-port": 3000,  
        "resources-limits-cpu": "100m",  
        "resources-limits-memory": "128Mi",  
        "resources-requests-cpu": "100m",  
        "resources-requests-memory": "128Mi",  
        "autoscaling-enabled": "",  
        "autoscaling-minReplicas": 1,  
        "autoscaling-maxReplicas": 2,  
        "autoscaling-metrics-cpu-targetAverageUtilization": "100m"  
      },  
      "outputs": []  
    }  
  ]  
}
```

```

}, {
  "publichelm/wordpress": {
    "title": "publichelm/wordpress",
    "template_name": "wordpress",
    "version": "0.8.8",
    "id": "publichea9b36841",
    "template_type": "Helm",
    "template_content_type": "Cloud Pak for Multicloud Management",
    "template_provider": "Cloud Pak for Multicloud Management",
    "instance_name": "",
    "cloud_connection_name": "MCM Connection",
    "template_dataobjects": {},
    "template_params": {
      "namespace": "services",
      "image": "bitnami/wordpress:4.9.4-r2",
      "imagePullPolicy": "IfNotPresent",
      "wordpressUsername": "user",
      "wordpressEmail": "user@example.com",
      "wordpressFirstName": "FirstName",
      "wordpressLastName": "LastName",
      "wordpressBlogName": "User's Blog!",
      "allowEmptyPassword": "yes",
      "externalDatabase-host": "localhost",
      "externalDatabase-user": "bn_wordpress",
      "externalDatabase-password": "",
      "externalDatabase-database": "bitnami_wordpress",
      "externalDatabase-port": 3306,
      "mariadb-enabled": true,
      "mariadb-mariadbDatabase": "bitnami_wordpress",
      "mariadb-mariadbUser": "bn_wordpress",
      "mariadb-persistence-enabled": true,
      "mariadb-persistence-accessMode": "ReadWriteOnce",
      "mariadb-persistence-size": "8Gi",
      "serviceType": "LoadBalancer",
      "healthcheckHttps": "",
      "livenessProbe-initialDelaySeconds": 120,
      "livenessProbe-periodSeconds": 10,
      "livenessProbe-timeoutSeconds": 5,
      "livenessProbe-failureThreshold": 6,
      "livenessProbe-successThreshold": 1,
      "readinessProbe-initialDelaySeconds": 30,
      "readinessProbe-periodSeconds": 10,
      "readinessProbe-timeoutSeconds": 5,
      "readinessProbe-failureThreshold": 6,
      "readinessProbe-successThreshold": 1,
      "ingress-enabled": "",
      "ingress-hosts-0-name": "wordpress.local",
      "ingress-hosts-0-tls": "",
      "ingress-hosts-0-tlsSecret": "wordpress.local-
tls",
      "ingress-hosts-0-annotations": "",
      "ingress-secrets": "",
      "persistence-enabled": true,
      "persistence-accessMode": "ReadWriteOnce",
      "persistence-size": "10Gi",
      "resources-requests-memory": "512Mi",
      "resources-requests-cpu": "300m"
    },
    "outputs": []
  }, {
    "publichelm/filebeat": {
      "title": "publichelm/filebeat",
      "template_name": "filebeat",

```

```

        "version": "0.1.0",
        "id": "publiche046d1a27",
        "template_type": "Helm",
        "template_content_type": "Cloud Pak for Multicloud
Management",
        "template_provider": "Cloud Pak for Multicloud
Management",
        "instance_name": "",
        "cloud_connection_name": "MCM Connection",
        "template_dataobjects": {},
        "template_params": {
            "namespace": "kube-system",
            "image-repository":
"docker.elastic.co/beats/filebeat",
            "image-tag": "6.2.2",
            "image-pullPolicy": "IfNotPresent",
            "config-processors-0-add_cloud_metadata": "",
            "rbac-create": true,
            "serviceAccount-create": true,
            "serviceAccount-name": "",
            "config-filebeat.config-prospectors-path":
"${path.config}/prospectors.d/*.yaml",
            "config-filebeat.config-prospectors-
reload.enabled": "",
            "config-filebeat.config-modules-path":
"${path.config}/modules.d/*.yaml",
            "config-filebeat.config-modules-reload.enabled":
"",
            "config-filebeat.prospectors-0-type": "log",
            "config-filebeat.prospectors-0-enabled": true,
            "config-filebeat.prospectors-0-paths-0":
"/var/log/**",
            "config-filebeat.prospectors-0-exclude_files-0": "
(\\.[0-9]{1,})$",
            "config-filebeat.prospectors-0-exclude_files-1":
"templates/clusterrolebinding.yaml.gz$",
            "config-output.file-path":
"/usr/share/filebeat/data",
            "config-output.file-filename": "filebeat",
            "config-output.file-rotate_every_kb": 10000,
            "config-output.file-number_of_files": 5
        },
        "outputs": []
    }
},
    "tag": "5a8e69872210e0001d311d9a",
    "template_mapping": {
        "ibmchartc99c9c4a": "ibm-nodejs-sample",
        "publichea9b36841": "wordpress",
        "publiche046d1a27": "filebeat"
    },
    "id": "5ab0dcb27bf798001e85724a",
    "orgGuid": "dummy-org-id",
    "spaceGuid": "dummy-space-id",
    "namespaceId": "dummy-space-id",
    "displayName": "helmmultiple"
}
]
]

```

Get Service versions for a service by using service-id

This API fetches all the service versions for a specific service by using the service ID.


```
curl -k -X GET \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICEID}/ServiceVersions?
filter[order]=updated_time%20DESC&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAME
SPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
```

Response:

```
[{"version":"1.0.0.0","ServiceEngine_InputFile":{"service":
{"specVersion":"v3","catalog_metadata":
{"name":"T_1594640403754979b9fb0_c4fd_11ea_9354_f721359f251a_9be04e40-c4fd-11ea-
9354-f721359f251a","description":"","image":"serviceicon_1.svg","category":"CloudServic
es","bullets":[{"title":"Cloud Provider","description":"IBM Bluemix
Platform"}],"providerDisplayName":"IBM","longDescription":"Meant Stack deployed on
Multiple VMs to help user get the stack working is faster
manner","documentationUrl":"http://example.com/meanstack","supportUrl":"http://exa
mple.com/meanstack","bindable":"","updatable":"","systemTags":true},"tags":
[],"actions":[{"type":"provision","name":"Provision","configuration":
{"onError":"pause"},"input_parameters":[],"flow":{"conditions":[],"templates":
[],"resthooks":[{"resthook":
{"title":"resthook_d4961d","id":"resthook7afd540e","headers":
{"accept":"application/json","Content-
Type":"application/json"},"url":"https://jsonplaceholder.typicode.com/todos/1","me
thod":"GET"}]},"notifications":[],"sequence":
{"0":"resthook7afd540e"},"error_sequence":{"},"abort_sequence":
{},"output_parameters":[]},"plans":[{"name":"Standard","description":"To deploy
a Standard plan","actions":[],"plan_parameters":
[]},"version":"1.0.0.0"}],"service_id":"5f0c481b415e1137d7efc5f9","namespaceId":
"ibmcom","author":"user2_administrator","specVersion":"v3","status":"Published","pu
blishedTime":"2020-08-25T14:29:47.231Z","tags":
[{"name":"service_identifier","label":"service_identifier","type":"string","immuta
ble":true,"hidden":false,"required":false,"secured":false,"default":"${svc_instanc
e.service_instance_id}","customtype":"string","permission":"Read-
Only","description":"Unique identifier generated by ICAM which is mapped to the
service\n instance of the template provisioned.","isSystemTag":true},
{"name":"service_name","label":"service_name","type":"string","immutable":true,"hi
dden":false,"required":false,"secured":false,"default":"${svc_instance.name}","cus
tomtype":"string","permission":"Read-Only","description":"Name provided for the
service instance by the end user at time of request","isSystemTag":true},
{"name":"request_user","label":"request_user","type":"string","immutable":true,"hi
dden":false,"required":false,"secured":false,"default":"${svc_instance.owner}","cu
stomtype":"string","permission":"Read-Only","description":"The user id of the
current user that requested the provisioning of the
template.","isSystemTag":true},
{"name":"request_group","label":"request_group","type":"string","immutable":true,"
hidden":false,"required":false,"secured":false,"default":"${svc_instance.group}","
customtype":"string","permission":"Read-Only","description":"The current context
group id of the current user that requested\n the provisioning of the template.
This is important because a user may be part of\n more than one group and this
identifies the context of which group the user made the
request.","isSystemTag":true}], "systemTags":true,"creationTime":"2020-07-
13T11:40:11.292Z","lastUpdated":"2020-08-
25T14:29:47.231Z","serviceInstanceCount":3,"id":"5f0c481b415e1137d7efc5fa","action
s":[{"type":"provision","name":"Provision","configuration":
{"onError":"pause"},"input_parameters":[],"flow":{"conditions":[],"templates":
[],"resthooks":[{"resthook":
{"title":"resthook_d4961d","id":"resthook7afd540e","headers":
{"accept":"application/json","Content-
Type":"application/json"},"url":"https://jsonplaceholder.typicode.com/todos/1","me
thod":"GET"}]},"notifications":[],"sequence":
{"0":"resthook7afd540e"},"error_sequence":{"},"abort_sequence":
```

```

{}}, "output_parameters": []}, "plans": [{"name": "Standard", "description": "To deploy
a Standard plan", "actions": [], "plan_parameters":
[]}], "tagId": "5ecddfe55d6b7600155831de", "retiredTime": "2020-08-25T14:29:31.229Z"},
{"version": "2.0.0.0", "ServiceEngine_InputFile": {"service":
{"specVersion": "v3", "catalog_metadata":
{"name": "T_1594640403754979b9fb0_c4fd_11ea_9354_f721359f251a_9be04e40-c4fd-11ea-
9354-f721359f251a", "description": "test-with-resthook
desc", "image": "serviceicon_1.svg", "category": "CloudServices", "bullets":
[], "providerDisplayName": "IBM", "longDescription": "test-with-resthook long
desc", "documentationUrl": "", "supportUrl": "", "bindable": false, "updatable": "", "systemTags": true, "author": "admin"}, "tags":
[{"name": "request_group", "label": "request_group", "type": "string", "immutable": true,
"hidden": false, "required": false, "secured": false, "description": "The current context
group id of the current user that requested the provisioning of the template. This
is important because a user may be part of more than one group and this identifies
the context of which group the user made the
request.", "isSystemTag": true, "default": "${svc_instance.group}", "customtype": "string", "permission": "Read-Only"},
{"name": "request_user", "label": "request_user", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The user id of the
current user that requested the provisioning of the
template.", "isSystemTag": true, "default": "${svc_instance.owner}", "customtype": "string", "permission": "Read-Only"},
{"name": "service_name", "label": "service_name", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Name provided for the
service instance by the end user at time of
request", "isSystemTag": true, "default": "${svc_instance.name}", "customtype": "string", "permission": "Read-Only"},
{"name": "service_identifier", "label": "service_identifier", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Unique
identifier generated by ICAM which is mapped to the service instance of the
template
provisioned.", "isSystemTag": true, "default": "${svc_instance.service_instance_id}", "customtype": "string", "permission": "Read-Only"}]}, "actions":
[{"type": "provision", "name": "Provision", "description": "Default action for
deployment of service", "input_parameters":
[{"name": "arrayparam", "label": "arrayparam", "customtype": "list", "type": "list", "immutable": false, "hidden": false, "required": false, "secured": false, "default":
["first", "second", "third"], "editable": true}], "flow": {"conditions": [], "templates":
[]}, "resthooks": [{"resthook":
{"title": "resthook_fb4ca7", "id": "resthook1b74da50", "headers":
{"accept": "application/json", "Content-Type": "application/json"}, "url": "http://9.5.28.173:3000/posts", "method": "POST", "payload":
{"id": 2001, "arraylistcustom": "${input_parameters.arrayparam}"}]}]}, "notifications":
[], "sequence": {"0": "resthook1b74da50"}, "error_sequence": {}, "output_parameters":
[]}], "plans": [{"name": "Standard", "description": "To deploy a Standard
plan", "actions": [], "plan_parameters":
[]}], "version": "2.0.0.0"}}, {"service_id": "5f0c481b415e1137d7efc5f9", "tenantId": "72e
1331d-3d93-447a-bf61-
de1177f2f0d8", "namespaceId": "ibmcom", "author": "admin", "specVersion": "v3", "status":
"Published", "publishedTime": "2020-07-13T11:40:18.421Z", "tags":
[{"name": "request_group", "label": "request_group", "type": "string", "immutable": true,
"hidden": false, "required": false, "secured": false, "description": "The current context
group id of the current user that requested the provisioning of the template. This
is important because a user may be part of more than one group and this identifies
the context of which group the user made the
request.", "isSystemTag": true, "default": "${svc_instance.group}", "customtype": "string", "permission": "Read-Only"},
{"name": "request_user", "label": "request_user", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The user id of the
current user that requested the provisioning of the
template.", "isSystemTag": true, "default": "${svc_instance.owner}", "customtype": "string", "permission": "Read-Only"}],

```

```
{
  "name": "service_name", "label": "service_name", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Name provided for the service instance by the end user at time of request", "isSystemTag": true, "default": "${svc_instance.name}", "customtype": "string", "permission": "Read-Only",
  "name": "service_identifier", "label": "service_identifier", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Unique identifier generated by ICAM which is mapped to the service instance of the template provisioned.", "isSystemTag": true, "default": "${svc_instance.service_instance_id}", "customtype": "string", "permission": "Read-Only"
}, {
  "systemTags": true, "creationTime": "2020-07-13T11:40:18.186Z", "lastUpdated": "2020-07-13T11:40:18.421Z", "serviceInstanceCount": 1, "id": "5f0c4822415e1137d7efc5ff", "actions": [
    {
      "type": "provision", "name": "Provision", "description": "Default action for deployment of service", "input_parameters": [
        {
          "name": "arrayparam", "label": "arrayparam", "customtype": "list", "type": "list", "immutable": false, "hidden": false, "required": false, "secured": false, "default": ["first", "second", "third"], "editable": true
        }
      ], "flow": {
        "conditions": [], "templates": [], "resthooks": [
          {
            "resthook": {
              "title": "resthook_fb4ca7", "id": "resthook1b74da50", "headers": {
                "accept": "application/json", "Content-Type": "application/json"
              }, "url": "http://9.5.28.173:3000/posts", "method": "POST", "payload": {
                "id": 2001, "arraylistcustom": "${input_parameters.arrayparam}"
              }
            }
          }
        ], "notifications": [
          {
            "sequence": {
              "0": "resthook1b74da50", "error_sequence": {}
            }, "output_parameters": [
              {
                "name": "Standard", "description": "To deploy a Standard plan"
              }
            ], "actions": [
              {
                "plan_parameters": [
                  {
                    "tagId": "5ecddfe55d6b7600155831de"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ]
}
```

Get Service version manifest by service version ID

This API fetches the service version manifest by using the service version ID.

```
curl -k -X GET \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_VERSION_ID}?ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
```

Response:

```
{
  "version": "1.1.1.1", "ServiceEngine_InputFile": {
    "service": {
      "version": "1.1.1.1", "specVersion": "v3", "catalog_metadata": {
        "name": "Test_Automationb7550ee0-b6a7-11ea-9f4e-f3a3b131b5d4", "description": "", "image": "serviceicon_1.svg", "category": "CloudServices", "bullets": [
          {
            "providerDisplayName": "IBM", "longDescription": "", "documentationUrl": "", "supportUrl": "", "bindable": false, "updatable": "", "systemTags": true
          }
        ], "tags": [
          {
            "name": "request_group", "label": "request_group", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.", "isSystemTag": true, "default": "${svc_instance.group}", "customtype": "string", "permission": "Read-Only",
            "name": "request_user", "label": "request_user", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The user id of the current user that requested the provisioning of the template.", "isSystemTag": true, "default": "${svc_instance.owner}", "customtype": "string", "permission": "Read-Only",
            "name": "service_name", "label": "service_name", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Name provided for the service instance by the end user at time of request", "isSystemTag": true, "default": "${svc_instance.name}", "customtype": "string", "permission": "Read-Only"
          }
        ]
      }
    }
  }
}
```

```

    "dden":false,"required":false,"secured":false,"description":"Name provided for the
    service instance by the end user at time of
    request","isSystemTag":true,"default":"${svc_instance.name}","customtype":"string"
    ,"permission":"Read-Only"},
    {"name":"service_identifier","label":"service_identifier","type":"string","immutab
    le":true,"hidden":false,"required":false,"secured":false,"description":"Unique
    identifier generated by ICAM which is mapped to the service instance of the
    template
    provisioned.","isSystemTag":true,"default":"${svc_instance.service_instance_id}","
    customtype":"string","permission":"Read-Only"},"actions":
    [{"type":"provision","name":"Provision","description":"Default action for
    deployment of service","input_parameters":[{"name":"url_1","label":"URL
    1","customtype":"string","type":"string","immutable":false,"hidden":false,"require
    d":false,"secured":false,"isDynamic":false,"editable":true},
    {"name":"url_2","label":"URL
    2","customtype":"string","type":"string","immutable":false,"hidden":false,"require
    d":false,"secured":false,"description":"","isDynamic":false,"editable":true}],"flo
    w":{"conditions":[],"templates":[],"resthooks":[{"resthook":
    {"title":"resthook_5477d2","id":"resthooke76554d","headers":
    {"accept":"application/json","Content-
    Type":"application/json"},"immutable":false,"url":"${input_parameters.url_1}","met
    hod":"GET"}},{"resthook":
    {"title":"resthook_f2fce4","id":"resthook4712f3c3","headers":
    {"accept":"application/json","Content-
    Type":"application/json"},"immutable":false,"url":"${input_parameters.url_2}","met
    hod":"GET"}]}],"notifications":[],"sequence":
    {"0":"resthooke76554d","1":"resthook4712f3c3"},"error_sequence":
    {}},"output_parameters":[],"plans":[{"name":"Standard","description":"To deploy
    a Standard plan","actions":[],"plan_parameters":
    []}]}},"service_id":"5ef43b0ccfd9b035e2a3824b","namespaceId":"cert-
    manager","author":"user1_administrator","specVersion":"v3","status":"Published","p
    ublishedTime":"2020-06-25T05:50:06.602Z","tags":
    [{"name":"request_group","label":"request_group","type":"string","immutable":true,
    "hidden":false,"required":false,"secured":false,"description":"The current context
    group id of the current user that requested the provisioning of the template. This
    is important because a user may be part of more than one group and this identifies
    the context of which group the user made the
    request.","isSystemTag":true,"default":"${svc_instance.group}","customtype":"strin
    g","permission":"Read-Only"},
    {"name":"request_user","label":"request_user","type":"string","immutable":true,"hi
    dden":false,"required":false,"secured":false,"description":"The user id of the
    current user that requested the provisioning of the
    template.","isSystemTag":true,"default":"${svc_instance.owner}","customtype":"stri
    ng","permission":"Read-Only"},
    {"name":"service_name","label":"service_name","type":"string","immutable":true,"hi
    dden":false,"required":false,"secured":false,"description":"Name provided for the
    service instance by the end user at time of
    request","isSystemTag":true,"default":"${svc_instance.name}","customtype":"string"
    ,"permission":"Read-Only"},
    {"name":"service_identifier","label":"service_identifier","type":"string","immutab
    le":true,"hidden":false,"required":false,"secured":false,"description":"Unique
    identifier generated by ICAM which is mapped to the service instance of the
    template
    provisioned.","isSystemTag":true,"default":"${svc_instance.service_instance_id}","
    customtype":"string","permission":"Read-
    Only"},"systemTags":true,"creationTime":"2020-06-
    25T05:50:04.184Z","lastUpdated":"2020-06-
    25T05:50:06.602Z","serviceInstanceCount":0,"id":"5ef43b0ccfd9b035e2a3824c","action
    s":[{"type":"provision","name":"Provision","description":"Default action for
    deployment of service","input_parameters":[{"name":"url_1","label":"URL
    1","customtype":"string","type":"string","immutable":false,"hidden":false,"require
    d":false,"secured":false,"isDynamic":false,"editable":true},
    {"name":"url_2","label":"URL
    2","customtype":"string","type":"string","immutable":false,"hidden":false,"require

```

```
d":false,"secured":false,"description":"","isDynamic":false,"editable":true}], "flow": {"conditions": [], "templates": [], "resthooks": [{"resthook": {"title": "resthook_5477d2", "id": "resthooke76554d", "headers": {"accept": "application/json", "Content-Type": "application/json"}, "immutable": false, "url": "${input_parameters.url_1}", "method": "GET"}}, {"resthook": {"title": "resthook_f2fce4", "id": "resthook4712f3c3", "headers": {"accept": "application/json", "Content-Type": "application/json"}, "immutable": false, "url": "${input_parameters.url_2}", "method": "GET"}}], "notifications": [], "sequence": {"0": "resthooke76554d", "1": "resthook4712f3c3"}, "error_sequence": {}}, {"output_parameters": []}], "plans": [{"name": "Standard", "description": "To deploy a Standard plan", "actions": [], "plan_parameters": []}], "tagId": "5ecddfe55d6b7600155831de", "retiredTime": "2020-06-25T05:50:06.277Z", "isParametersResolved": false}
```

Publish a service version

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_MANIFEST_ID}/publish
?ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
```

Response:

```
{"version": "1.0.0.0", "ServiceEngine_InputFile": {"service": {"specVersion": "v3", "version": "1.0.0.0", "catalog_metadata": {"name": "Test_with_COMPATIBILITY_MODE_ON_16331c60-ed01-11ea-b709-43ed3173756c", "description": "", "image": "serviceicon_1.svg", "category": "CloudServices", "bullets": [], "providerDisplayName": "IBM", "longDescription": "", "documentationUrl": "", "supportUrl": "", "bindable": false, "updatable": "", "systemTags": true}, "tags": [{"name": "request_group", "label": "request_group", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.", "isSystemTag": true, "default": "${svc_instance.group}", "customtype": "string", "permission": "Read-Only"}, {"name": "request_user", "label": "request_user", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The user id of the current user that requested the provisioning of the template.", "isSystemTag": true, "default": "${svc_instance.owner}", "customtype": "string", "permission": "Read-Only"}, {"name": "service_name", "label": "service_name", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Name provided for the service instance by the end user at time of request", "isSystemTag": true, "default": "${svc_instance.name}", "customtype": "string", "permission": "Read-Only"}, {"name": "service_identifier", "label": "service_identifier", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Unique identifier generated by ICAM which is mapped to the service instance of the template provisioned.", "isSystemTag": true, "default": "${svc_instance.service_instance_id}", "customtype": "string", "permission": "Read-Only"}], "actions": [{"type": "provision", "name": "Provision", "description": "Default action for deployment of service", "input_parameters": [{"name": "vm-name", "label": "vm-name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1", "edi
```

```

table":true},
{"name":"Conn1","label":"Conn1","customtype":"connection","type":"connection","immutable":false,"hidden":false,"required":true,"secured":false,"isDynamic":false,"default":"vmware","provider":"VMware vSphere","editable":true},
{"name":"DataObject1","label":"DataObject1","customtype":"sharedparameter","type":"sharedparameter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"DefaultNoBastionHostRequired","datatype":"bastionhost","editable":true},
{"name":"region","label":"region","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"isDynamic":false,"default":"west1","editable":true},
{"name":"org","label":"Org","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"description":"","isDynamic":false,"default":"all","editable":true},
{"name":"team","label":"Team","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"description":"","isDynamic":false,"default":"team1","editable":true},
{"name":"DataObject2","label":"DataObject2","customtype":"sharedparameter","type":"sharedparameter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"","datatype":"httpproxy","editable":true},
{"name":"aws","label":"aws","customtype":"connection","type":"connection","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"","provider":"Amazon EC2","editable":true},
{"name":"size","label":"size","customtype":"counter","type":"counter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":100,"range":{"min":1,"max":1000},"editable":true},"flow":
{"conditions":[],"templates":[],"resthooks":[{"resthook":
{"title":"Prov_R1","id":"resthooka4f9e22b","headers":
{"accept":"application/json","Content-Type":"application/json"},"error":false,"immutable":false,"url":"http://9.30.167.199:8080/pingpong/list/1?id=1021&list=${input_parameters.Conn1}&list=${input_parameters.DataObject1}","method":"GET","warning":false}}],"notifications":[],"sequence":
{"0":"resthooka4f9e22b"},"error_sequence":{},"output_parameters":[]},"plans":
[{"name":"Standard","description":"To deploy a Standard plan","actions":
[],"plan_parameters":
[{"name":"DataObject2","label":"DataObject2","customtype":"sharedparameter","type":"sharedparameter","immutable":true,"hidden":true,"required":true,"secured":false,"description":"","isDynamic":false,"default":"DefaultNoProxyRequired","datatype":"httpproxy"},
{"name":"aws","label":"aws","customtype":"connection","type":"connection","immutable":true,"hidden":true,"required":true,"secured":false,"description":"","isDynamic":false,"default":"AWS","provider":"Amazon EC2"},
{"name":"size","label":"size","customtype":"counter","type":"counter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":50,"range":{"min":10,"max":100}},{"name":"vm-name","label":"vm-name","customtype":"string","type":"string","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"vm1"},
{"name":"Conn1","label":"Conn1","customtype":"connection","type":"connection","immutable":false,"hidden":false,"required":true,"secured":false,"isDynamic":false,"default":"vmware","provider":"VMware vSphere"},
{"name":"DataObject1","label":"DataObject1","customtype":"sharedparameter","type":"sharedparameter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"DefaultNoBastionHostRequired","datatype":"bastionhost"},
{"name":"region","label":"region","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"isDynamic":false,"default":"west1"},
{"name":"org","label":"Org","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"description":"","isDynamic":false,"default":"all"},
{"name":"team","label":"Team","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"description":"","isDynamic":f

```

```

    else, "default": "team1" ] ] ] }, { "name": "Max", "description": "Max", "plan_parameters":
    [ { "name": "size", "label": "size", "customtype": "counter", "type": "counter", "immutable":
    : false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic"
    : false, "default": 100, "range": { "min": 100, "max": 1000 } }, { "name": "vm-
    name", "label": "vm-
    name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "requ
    ired": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1" },
    { "name": "Conn1", "label": "Conn1", "customtype": "connection", "type": "connection", "imm
    utable": false, "hidden": false, "required": true, "secured": false, "isDynamic": false, "de
    fault": "vmware", "provider": "VMware vSphere" },
    { "name": "DataObject1", "label": "DataObject1", "customtype": "sharedparameter", "type":
    "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false
    , "description": "", "isDynamic": false, "default": "DefaultNoBastionHostRequired", "data
    type": "bastionhost" },
    { "name": "region", "label": "region", "customtype": "string", "type": "string", "immutable
    ": false, "hidden": false, "required": false, "secured": false, "isDynamic": false, "default
    ": "west1" },
    { "name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": fals
    e, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": fal
    se, "default": "all" },
    { "name": "team", "label": "Team", "customtype": "string", "type": "string", "immutable": fa
    lse, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": f
    alse, "default": "team1" },
    { "name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type":
    "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false
    , "description": "", "isDynamic": false, "default": "", "datatype": "httpproxy" },
    { "name": "aws", "label": "aws", "customtype": "connection", "type": "connection", "immutab
    le": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynam
    ic": false, "default": "", "provider": "Amazon EC2" } ] ] },
    { "name": "Min", "description": "Min", "plan_parameters":
    [ { "name": "region", "label": "region", "customtype": "string", "type": "string", "immutabl
    e": true, "hidden": true, "required": false, "secured": false, "isDynamic": false, "default
    ": "west1" },
    { "name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": true
    , "hidden": true, "required": false, "secured": false, "description": "", "isDynamic": false
    , "default": "all" },
    { "name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type":
    "sharedparameter", "immutable": true, "hidden": true, "required": true, "secured": false, "
    description": "", "isDynamic": false, "default": "DefaultNoProxyRequired", "datatype": "h
    ttpproxy" },
    { "name": "aws", "label": "aws", "customtype": "connection", "type": "connection", "immutab
    le": true, "hidden": true, "required": true, "secured": false, "description": "", "isDynamic
    ": false, "default": "AWS", "provider": "Amazon EC2" },
    { "name": "size", "label": "size", "customtype": "counter", "type": "counter", "immutable":
    false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic":
    false, "default": 1, "range": { "min": 1, "max": 10 } }, { "name": "vm-name", "label": "vm-
    name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "requ
    ired": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1" },
    { "name": "Conn1", "label": "Conn1", "customtype": "connection", "type": "connection", "imm
    utable": false, "hidden": false, "required": true, "secured": false, "isDynamic": false, "de
    fault": "vmware", "provider": "VMware vSphere" },
    { "name": "DataObject1", "label": "DataObject1", "customtype": "sharedparameter", "type":
    "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false
    , "description": "", "isDynamic": false, "default": "DefaultNoBastionHostRequired", "data
    type": "bastionhost" },
    { "name": "team", "label": "Team", "customtype": "string", "type": "string", "immutable": fa
    lse, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": f
    alse, "default": "team1" ] ] ] }, { "service_id": "5f4f69cf643bcb0f7b35ab29", "tenantId": "7
    2e1331d-3d93-447a-bf61-del177f2f0d8", "namespaceId": "cert-
    manager", "author": "admin", "specVersion": "v3", "status": "Draft", "tags":
    [ { "name": "request_group", "label": "request_group", "type": "string", "immutable": true,
    "hidden": false, "required": false, "secured": false, "description": "The current context
    group id of the current user that requested the provisioning of the template. This
    is important because a user may be part of more than one group and this identifies

```

```

the context of which group the user made the
request.", "isSystemTag": true, "default": "${svc_instance.group}", "customtype": "string", "permission": "Read-Only"},
{"name": "request_user", "label": "request_user", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "The user id of the current user that requested the provisioning of the template.", "isSystemTag": true, "default": "${svc_instance.owner}", "customtype": "string", "permission": "Read-Only"},
{"name": "service_name", "label": "service_name", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Name provided for the service instance by the end user at time of request", "isSystemTag": true, "default": "${svc_instance.name}", "customtype": "string", "permission": "Read-Only"},
{"name": "service_identifier", "label": "service_identifier", "type": "string", "immutable": true, "hidden": false, "required": false, "secured": false, "description": "Unique identifier generated by ICAM which is mapped to the service instance of the template provisioned.", "isSystemTag": true, "default": "${svc_instance.service_instance_id}", "customtype": "string", "permission": "Read-Only"}], "systemTags": true, "creationTime": "2020-09-02T09:45:51.304Z", "serviceInstanceCount": 0, "id": "5f4f69cf643bcb0f7b35ab2a", "actions": [{"type": "provision", "name": "Provision", "description": "Default action for deployment of service", "input_parameters": [{"name": "vm-name", "label": "vm-name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1", "editable": true}, {"name": "Conn1", "label": "Conn1", "customtype": "connection", "type": "connection", "immutable": false, "hidden": false, "required": true, "secured": false, "isDynamic": false, "default": "vmware", "provider": "VMware vSphere", "editable": true}, {"name": "DataObject1", "label": "DataObject1", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "DefaultNoBastionHostRequired", "datatype": "bastionhost", "editable": true}, {"name": "region", "label": "region", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "isDynamic": false, "default": "west1", "editable": true}, {"name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "all", "editable": true}, {"name": "team", "label": "Team", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "team1", "editable": true}, {"name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "", "datatype": "httpproxy", "editable": true}, {"name": "aws", "label": "aws", "customtype": "connection", "type": "connection", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "", "provider": "Amazon EC2", "editable": true}, {"name": "size", "label": "size", "customtype": "counter", "type": "counter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": 100, "range": {"min": 1, "max": 1000}, "editable": true}], "flow": {"conditions": [], "templates": [], "resthooks": [{"resthook": {"title": "Prov_R1", "id": "resthooka4f9e22b", "headers": {"accept": "application/json", "Content-Type": "application/json"}, "error": false, "immutable": false, "url": "http://9.30.167.199:8080/pingpong/list/1?id=1021&list=${input_parameters.Conn1}&list=${input_parameters.DataObject1}", "method": "GET", "warning": false}}, {"notifications": [], "sequence": {"0": "resthooka4f9e22b"}, "error_sequence": {}, "output_parameters": []}], "plans": [{"name": "Standard", "description": "To deploy a Standard plan", "actions": [], "plan_parameters": [{"name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": true, "hidden": true, "required": true, "secured": false,

```



```

"description": "", "isDynamic": false, "default": "DefaultNoProxyRequired", "datatype": "
httpproxy"},
{"name": "aws", "label": "aws", "customtype": "connection", "type": "connection", "immutable": true, "hidden": true, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "AWS", "provider": "Amazon EC2"},
{"name": "size", "label": "size", "customtype": "counter", "type": "counter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": 50, "range": {"min": 10, "max": 100}}, {"name": "vm-name", "label": "vm-name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1"},
{"name": "Conn1", "label": "Conn1", "customtype": "connection", "type": "connection", "immutable": false, "hidden": false, "required": true, "secured": false, "isDynamic": false, "default": "vmware", "provider": "VMware vSphere"},
{"name": "DataObject1", "label": "DataObject1", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "DefaultNoBastionHostRequired", "datatype": "bastionhost"},
{"name": "region", "label": "region", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "isDynamic": false, "default": "west1"},
{"name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "all"},
{"name": "team", "label": "Team", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "team1"}], {"name": "Max", "description": "Max", "plan_parameters": [{"name": "size", "label": "size", "customtype": "counter", "type": "counter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": 100, "range": {"min": 100, "max": 1000}}, {"name": "vm-name", "label": "vm-name", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "vm1"}, {"name": "Conn1", "label": "Conn1", "customtype": "connection", "type": "connection", "immutable": false, "hidden": false, "required": true, "secured": false, "isDynamic": false, "default": "vmware", "provider": "VMware vSphere"}, {"name": "DataObject1", "label": "DataObject1", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "DefaultNoBastionHostRequired", "datatype": "bastionhost"}, {"name": "region", "label": "region", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "isDynamic": false, "default": "west1"}, {"name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "all"}, {"name": "team", "label": "Team", "customtype": "string", "type": "string", "immutable": false, "hidden": false, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "team1"}, {"name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "", "datatype": "httpproxy"}, {"name": "aws", "label": "aws", "customtype": "connection", "type": "connection", "immutable": false, "hidden": false, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "", "provider": "Amazon EC2"}], {"name": "Min", "description": "Min", "plan_parameters": [{"name": "region", "label": "region", "customtype": "string", "type": "string", "immutable": true, "hidden": true, "required": false, "secured": false, "isDynamic": false, "default": "west1"}, {"name": "org", "label": "Org", "customtype": "string", "type": "string", "immutable": true, "hidden": true, "required": false, "secured": false, "description": "", "isDynamic": false, "default": "all"}, {"name": "DataObject2", "label": "DataObject2", "customtype": "sharedparameter", "type": "sharedparameter", "immutable": true, "hidden": true, "required": true, "secured": false, "description": "", "isDynamic": false, "default": "DefaultNoProxyRequired", "datatype": "h

```

```

tttproxy"},
{"name":"aws","label":"aws","customtype":"connection","type":"connection","immutable":true,"hidden":true,"required":true,"secured":false,"description":"","isDynamic":false,"default":"AWS","provider":"Amazon EC2"},
{"name":"size","label":"size","customtype":"counter","type":"counter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":1,"range":{"min":1,"max":10}},{"name":"vm-name","label":"vm-name","customtype":"string","type":"string","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"vm1"},
{"name":"Conn1","label":"Conn1","customtype":"connection","type":"connection","immutable":false,"hidden":false,"required":true,"secured":false,"isDynamic":false,"default":"vmware","provider":"VMware vSphere"},
{"name":"DataObject1","label":"DataObject1","customtype":"sharedparameter","type":"sharedparameter","immutable":false,"hidden":false,"required":true,"secured":false,"description":"","isDynamic":false,"default":"DefaultNoBastionHostRequired","datatype":"bastionhost"},
{"name":"team","label":"Team","customtype":"string","type":"string","immutable":false,"hidden":false,"required":false,"secured":false,"description":"","isDynamic":false,"default":"team1"}]]], "tagId":"5f476f9881758e190de1c44e", "warning": []}

```

Duplicate a service

Creates a copy of the Service with all its versions in a new Service.

```

curl -k -X POST \
'https://{{TSA_HOSTNAME}}/cam/composer/api/v1/Service/{{SERVICE_ID}}/duplicateService?
ace_orgGuid={{MCM_TEAM}}&cloudOE_spaceGuid={{NAMESPACE}}&tenantId={{TSA_TENANT_ID}}' \
-H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_duplicate.json

```

Payload

```

{
  "service_name": "{{NEW-SERVICE-NAME}}"
}

```

Response

```

{
  "id": "5df3aacea051bd001d1611cf",
  "name": "SampleV2",
  "author": "admin",
  "image": "serviceicon_1.svg",
  "status": "",
  "serviceManifestId": "5df3aacea051bd001d1611d0",
  "tag": "5c6e73077077fc001715703b",
  "creationTime": "2019-12-13T15:14:22.045Z",
  "lastUpdated": "2019-12-13T15:14:22.156Z",
  "namespaceId": "default",
  "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
  "description": "A Sample Single Virtual Machine Service",
  "longDescription": "",
  "defaultServiceVersion": {
    "id": "5df3aacea051bd001d1611d0",
    "version": "1.0.0.1",
    "ServiceEngine_InputFile": {
      "service": {
        "version": "1.0.0.1",
        "specVersion": "v3",

```

```

"catalog_metadata": {
  "name": "SampleV2",
  "description": "A Sample Single Virtual Machine Service",
  "image": "serviceicon_1.svg",
  "category": "CloudServices",
  "bullets": [],
  "providerDisplayName": "IBM",
  "longDescription": "",
  "documentationUrl": "",
  "supportUrl": "",
  "bindable": false,
  "updatable": "",
  "systemTags": true,
  "author": "admin"
},
"tags": [
  {
    "name": "request_group",
    "label": "request_group",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
    "isSystemTag": true,
    "default": "${svc_instance.group}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "request_user",
    "label": "request_user",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "default": "${svc_instance.owner}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user
at time of request",
    "isSystemTag": true,
    "default": "${svc_instance.name}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {

```

```

        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [
            {
                "name": "vmMemory",
                "label": "vmMemory",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            },
            {
                "name": "vmCpu",
                "label": "vmCpu",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            }
        ],
        "flow": {
            "conditions": [],
            "templates": [
                {
                    "SingleVirtualMachine": {
                        "title": "SingleVirtualMachine_cca453",
                        "template_name": "SingleVirtualMachine",
                        "version": "2.3",
                        "id": "singlevi23cd3149",
                        "template_type": "Terraform",
                        "template_content_type": "VMware vSphere",
                        "template_provider": "VMware vSphere",
                        "instance_name": "",
                        "cloud_connection_name": "vmware",
                        "template_data_objects": {},
                        "template_params": {
                            "vm_1_name": "ver2vm",
                            "vm_1_dns_servers": [
                                "1.1.1.2"
                            ],
                            "vm_1_dns_suffixes": [
                                "test.cam.com"
                            ]
                        }
                    }
                }
            ]
        }
    }
]

```

```

    ],
    "vm_1_root_disk_keep_on_remove": "false",
    "vm_1_domain": "test.cam.com",
    "vm_1-image": "terraform/template/rhel_7_2_small",
    "vm_1_root_disk_size": "25",
    "vm_1_root_disk_controller_type": "scsi",
    "vm_1_root_disk_datastore": "NSXcluster/datastore1",
    "vm_1_root_disk_type": "eager_zeroed",
    "vm_1_ipv4_gateway": "10.4.10.1",
    "vm_1_ipv4_address": "10.4.10.17",
    "vm_1_memory": "${input_parameters.vmMemory}",
    "vm_1_ipv4_prefix_length": "23",
    "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
    "vm_1_network_interface_label": "external-net-410",
    "vm_1_cluster": "icm-vmware-nsx-2",
    "vm_1_datacenter": "NSX Manage To",
    "vm_1_folder": "terraform/test",
    "vm_1_adapter_type": "vmxnet3",
    "vm_1_resource_pool": "cam"
  },
  "outputs": [],
  "immutable": false,
  "provider_engine_version": "0.11.11"
}
}
],
"resthooks": [],
"notifications": [
  {
    "emailnotification": {
      "title": "emailnotification_4278ce",
      "type": "email",
      "receiverList": "testmail@test-mail.sample.net",
      "id": "emailnot21410720",
      "sender": "CAMadmin",
      "copyToList": "",
      "subject": "VM Creation Succesful",
      "body": "VM create at
${templates.singlevi23cd3149.created_at}",
      "immutable": false
    }
  },
  {
    "emailnotification": {
      "title": "emailnotification_ac336e",
      "type": "email",
      "receiverList": "testmail@test-mail.sample.net",
      "id": "emailnotd384eeec",
      "sender": "CAMadmin",
      "copyToList": "",
      "subject": "VM Creation failed",
      "body": "VM creation failure caused by :
${svc_instance.error.message}",
      "immutable": false
    }
  }
],
"sequence": {
  "0": "singlevi23cd3149",
  "1": "emailnot21410720"
},
"error_sequence": {
  "0": "emailnotd384eeec"
}
}

```

```

        },
        "output_parameters": []
    }
],
"plans": [
    {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
    }
]
}
},
"service_id": "5df3aacea051bd001d1611cf",
"namespaceId": "default",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user at time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",

```

```

        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"systemTags": true,
"creationTime": "2019-12-13T15:14:22.062Z",
"serviceInstanceCount": 0,
},
"warning": []
}

```

Copy service version to a service

```

curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SOURCE_SERVICE_VERSION_ID}/copyService?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_version_copy.json

```

Payload

```

{
  "to_service_id": "${SERVICE_ID}",
  "new_version_name": "${NEW-VERSION-LABEL}"
}

```

Response

```

{
  "id": "5df3bb2fa051bd001d1611e1",
  "version": "1.0.0.2",
  "ServiceEngine_InputFile": {
    "service": {
      "version": "1.0.0.2",
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "SampleV2",
        "description": "A Sample Single Virtual Machine Service",
        "image": "serviceicon_1.svg",
        "category": "CloudServices",
        "bullets": [],
        "providerDisplayName": "IBM",
        "longDescription": "",
        "documentationUrl": "",
        "supportUrl": ""
      }
    }
  }
}

```

```

    "bindable": false,
    "updatable": "",
    "systemTags": true,
    "author": "admin"
  },
  "tags": [
    {
      "name": "request_group",
      "label": "request_group",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
      "isSystemTag": true,
      "default": "${svc_instance.group}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "request_user",
      "label": "request_user",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The user id of the current user that requested the
provisioning of the template.",
      "isSystemTag": true,
      "default": "${svc_instance.owner}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_name",
      "label": "service_name",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Name provided for the service instance by the end user
at time of request",
      "isSystemTag": true,
      "default": "${svc_instance.name}",
      "customtype": "string",
      "permission": "Read-Only"
    },
    {
      "name": "service_identifier",
      "label": "service_identifier",
      "type": "string",
      "immutable": true,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
      "isSystemTag": true,

```



```

        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [
            {
                "name": "vmMemory",
                "label": "vmMemory",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            },
            {
                "name": "vmCpu",
                "label": "vmCpu",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false
            }
        ],
        "flow": {
            "conditions": [],
            "templates": [
                {
                    "SingleVirtualMachine": {
                        "title": "SingleVirtualMachine_cca453",
                        "template_name": "SingleVirtualMachine",
                        "version": "2.3",
                        "id": "singlevi23cd3149",
                        "template_type": "Terraform",
                        "template_content_type": "VMware vSphere",
                        "template_provider": "VMware vSphere",
                        "instance_name": "",
                        "cloud_connection_name": "vmware",
                        "template_data_objects": {},
                        "template_params": {
                            "vm_1_name": "ver2vm",
                            "vm_1_dns_servers": [
                                "1.1.1.2"
                            ],
                            "vm_1_dns_suffixes": [
                                "test.cam.com"
                            ],
                            "vm_1_root_disk_keep_on_remove": "false",
                            "vm_1_domain": "test.cam.com",
                            "vm_1_image": "terraform/template/rhel_7_2_small",
                            "vm_1_root_disk_size": "25",
                            "vm_1_root_disk_controller_type": "scsi",
                            "vm_1_root_disk_datastore": "NSXcluster/datastore1",
                            "vm_1_root_disk_type": "eager_zeroed",
                            "vm_1_ipv4_gateway": "10.4.10.1",
                            "vm_1_ipv4_address": "10.4.10.17",

```

```

        "vm_1_memory": "${input_parameters.vmMemory}",
        "vm_1_ipv4_prefix_length": "23",
        "vm_1_number_of_vcpu": "${input_parameters.vmCpu}",
        "vm_1_network_interface_label": "external-net-410",
        "vm_1_cluster": "icm-vmware-nsx-2",
        "vm_1_datacenter": "NSX Manage To",
        "vm_1_folder": "terraform/test",
        "vm_1_adapter_type": "vmxnet3",
        "vm_1_resource_pool": "cam"
    },
    "outputs": [],
    "immutable": false
}
},
"resthooks": [],
"notifications": [
    {
        "emailnotification": {
            "title": "emailnotification_4278ce",
            "type": "email",
            "receiverList": "testmail@testr-mail.net",
            "id": "emailnot21410720",
            "sender": "CAMadmin",
            "copyToList": "",
            "subject": "VM Creation Succesful",
            "body": "VM create at ${templates.singlevi23cd3149.created_at}",
            "immutable": false
        }
    }
],
"sequence": {
    "0": "singlevi23cd3149",
    "1": "emailnot21410720"
},
"error_sequence": {}
},
"output_parameters": []
}
],
"plans": [
    {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
    }
]
}
},
"service_id": "5df3b815a051bd001d1611da",
"tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
"namespaceId": "default",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,

```

```

    "secured": false,
    "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
    "isSystemTag": true,
    "default": "${svc_instance.group}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "request_user",
    "label": "request_user",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "default": "${svc_instance.owner}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user at
time of request",
    "isSystemTag": true,
    "default": "${svc_instance.name}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
    "isSystemTag": true,
    "default": "${svc_instance.service_instance_id}",
    "customtype": "string",
    "permission": "Read-Only"
  }
],
"systemTags": true,
"creationTime": "2019-12-13T16:24:15.334Z",
"serviceInstanceCount": 0,
"warning": []
}

```

Change the category of a service

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/updateServiceDe
tails?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}'
\
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_change_category.json
```

Payload, content for service_change_category.json

```
{
  "service": {
    "catalog_metadata": {
      "category": "${NAME-OF-CATEGORY}"
    }
  }
}
```

Response

```
{
  "service": {
    "name": "SampleV1",
    "author": "admin",
    "image": "serviceicon_1.svg",
    "status": "",
    "serviceManifestId": "5df3aacea051bd001d1611d0",
    "tag": "5d8a3531bbcac2001d967d1d",
    "creationTime": "2019-12-13T15:14:22.045Z",
    "lastUpdated": "2019-12-13T15:30:09.054Z",
    "namespaceId": "default",
    "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
    "id": "5df3aacea051bd001d1611cf",
    "description": "A Sample Single Virtual Machine Service",
    "longDescription": ""
  }
}
```

Rename a service

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/updateServiceDe
tails?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}'
\
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_rename.json
```

Payload, content for service_rename.json

```
{
  "service": {
    "catalog_metadata": {
      "name": "SampleV3"
    }
  }
}
```

Response

```
{
  "service": {
    "id": "5df3aacea051bd001d1611cf",
    "name": "SampleV3",
    "author": "admin",
    "image": "serviceicon_1.svg",
    "serviceManifestId": "5df3aacea051bd001d1611d0",
    "tag": "5c6e73077077fc001715703b",
    "creationTime": "2019-12-13T15:14:22.045Z",
    "lastUpdated": "2019-12-13T15:22:38.185Z",
    "namespaceId": "default",
    "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
    "description": "A Sample Single Virtual Machine Service",
    "longDescription": ""
  }
}
```

Change service image for a service

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/${SERVICE_ID}/updateServiceDe
tails?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_change_image.json
```

Payload: content for service_change_image.json

```
{
  "service": {
    "catalog_metadata": {
      "image": "${NEW-IMAGE-NAME}"
    }
  }
}
```

Response

```
{
  "service": {
    "name": "SampleV1",
    "author": "admin",
    "image": "serviceicon_2.svg",
    "status": ""
    "serviceManifestId": "5df3aacea051bd001d1611d0",
    "tag": "5d8a3531bbcac2001d967d1d",
    "creationTime": "2019-12-13T15:14:22.045Z",
    "lastUpdated": "2019-12-13T15:30:09.054Z",
    "namespaceId": "default",
    "tenantId": "674a6a42-76c0-450c-a225-e743772c49f3",
    "id": "5df3aacea051bd001d1611cf",
    "description": "A Sample Single Virtual Machine Service",
    "longDescription": ""
  }
}
```

Assign access to a service

Assigns a single or multiple namespaces to a service.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Accept: application/json"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/Service/<id>/setNamespaceId?tenantId=
<tenantId>&cloudOE_spaceGuid=<comma separated list of namespaces>"
```

For scenarios of assigning access, see [Assigning access to a service](#)

Retire a service version

This API retires both published services and draft services.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_VERSION_ID}
/retire?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Unretire a service version

This API unretires the already retired service and moves it back to Published state.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_VERSION_ID}
/unretire?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Auto-retire a service version

This API retires a service version on the set date. The service version is retired at mid-night of the set day.

```
curl -k --location --request POST
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/${SERVICE_VERSION_ID}
/setRetireOnDate?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}' \
-H 'Accept: application/json' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-d @set_retire_on_date.json
```

set_retire_on_date.json:

```
{
  "retire_on_date": "2020-10-28T07:03:11.612Z"
}
```

Service Instance APIs

Deploy a service instance

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/{SERVICEMANIFEST_ID}/
deploy?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service_instance_create.json
```

service__instance_create.json:

```
{
  "name": "Service_Instance_Name",
  "instance_plan": "Standard",
  "instance_plan_id": "5dc13f03ec19fe001508e539",
  "tags_input": [
    {
      "value": "${svc_instance.group}",
      "label": "request_group",
      "name": "request_group",
      "secured": false
    },
    {
      "value": "${svc_instance.owner}",
      "label": "request_user",
      "name": "request_user",
      "secured": false
    },
    {
      "value": "${svc_instance.name}",
      "label": "service_name",
      "name": "service_name",
      "secured": false
    },
    {
      "value": "${svc_instance.service_instance_id}",
      "label": "service_identifier",
      "name": "service_identifier",
      "secured": false
    }
  ],
  "action": "Provision",
  "resourceProperties": {
    "templateParameters": [],
    "specifications": []
  },
  "ownership": {
    "username": "",
    "displayName": "admin",
    "orgGuid": "all",
    "spaceGuid": "abcdef123456",
    "tenantId": ""
  }
}
```

Response:

```
{
  "name": "Service_Instance_Name",
  "Status": "Submitted",
  "owner": "admin",
  "templateName": "",
  "ordered_time": "2020-08-25T06:15:42.245Z",
  "updated_time": "2020-08-25T06:15:42.246Z",
  "creationTime": "2020-08-25T06:15:42.245Z",
  "activity_deployment_details": {},
  "logs": "Waiting for service action to start",
  "serviceId": "5f0c481b415e1137d7efc5f9",
  "serviceManifestId": "5f0c481b415e1137d7efc5fa",
  "ownership": {
    "username": "admin",
    "displayName": "admin",
    "orgGuid": "all",
    "spaceGuid": "ibmcom",
    "tenantId": "4c46c3f2-6582-49de-b697-4891ae7e8dee",
    "namespaceId": "ibmcom",
    "tenantId": "4c46c3f2-6582-49de-b697-4891ae7e8dee",
    "instance_plan": "Standard",
    "action": "Provision",
    "id": "5f44ac8ee32082001cec758a",
    "ServiceID": "5f0c481b415e1137d7efc5f9",
    "instance_plan_id": "5f0c481b415e1137d7efc5fc",
    "resourceProperties": {
      "templateParameters": [],
      "specifications": []
    },
    "catalogItemId": "5f0c481b415e1137d7efc5f9",
    "tags_input":
  }
```

```
[{"value":"5f44ac8ee32082001cec758a","label":"service_identifier","name":"service_
identifier","secured":false},
{"value":"s9","label":"service_name","name":"service_name","secured":false},
{"value":"admin","label":"request_user","name":"request_user","secured":false},
{"value":"admin","label":"request_group","name":"request_group","secured":false}],
"ace_orgGuid":"all","cloudOE_spaceGuid":"ibmcom","Bearer_Token":"*****","bindabl
e":false,"catalogItemName":"T_1594640403754979b9fb0_c4fd_11ea_9354_f721359f251a_9b
e04e40-c4fd-11ea-9354-f721359f251a","serviceManifest":
{"version":"1.0.0.0","ServiceEngine_InputFile":{"service":
{"specVersion":"v3","catalog_metadata":
{"name":"T_1594640403754979b9fb0_c4fd_11ea_9354_f721359f251a_9be04e40-c4fd-11ea-
9354-
f721359f251a","description":"","image":"serviceicon_1.svg","category":"CloudServic
es","bullets":[{"title":"Cloud Provider","description":"IBM Bluemix
Platform"}],"providerDisplayName":"IBM","longDescription":"Meant Stack deployed on
Multiple VMs to help user get the stack working is faster
manner","documentationUrl":"http://example.com/meanstack","supportUrl":"http://exa
mple.com/meanstack","bindable":"","updatable":"","systemTags":true},"tags":
[],"actions":[{"type":"provision","name":"Provision","configuration":
{"onError":"pause"},"input_parameters":[],"flow":{"conditions":[],"templates":
[],"resthooks":[{"resthook":
{"title":"resthook_d4961d","id":"resthook7afd540e","headers":
{"accept":"application/json","Content-
Type":"application/json"},"url":"https://jsonplaceholder.typicode.com/todos/1","me
thod":"GET"}]},"notifications":[],"sequence":
{"0":"resthook7afd540e"},"error_sequence":{"},"abort_sequence":
{},"output_parameters":[],"plans":[{"name":"Standard","description":"To deploy
a Standard plan","actions":[],"plan_parameters":
[]},"version":"1.0.0.0"},"service_id":"5f0c481b415e1137d7efc5f9","namespaceId":
"ibmcom","author":"user2_administrator","specVersion":"v3","status":"Published","pu
blishedTime":"2020-07-13T11:40:17.901Z","tags":
[{"name":"service_identifier","label":"service_identifier","type":"string","immuta
ble":true,"hidden":false,"required":false,"secured":false,"default":"${svc_instanc
e.service_instance_id}","customtype":"string","permission":"Read-
Only","description":"Unique identifier generated by ICAM which is mapped to the
service\n instance of the template provisioned.","isSystemTag":true},
{"name":"service_name","label":"service_name","type":"string","immutable":true,"hi
dden":false,"required":false,"secured":false,"default":"${svc_instance.name}","cus
tomtype":"string","permission":"Read-Only","description":"Name provided for the
service instance by the end user at time of request","isSystemTag":true},
{"name":"request_user","label":"request_user","type":"string","immutable":true,"hi
dden":false,"required":false,"secured":false,"default":"${svc_instance.owner}","cu
stomtype":"string","permission":"Read-Only","description":"The user id of the
current user that requested the provisioning of the
template.","isSystemTag":true},
{"name":"request_group","label":"request_group","type":"string","immutable":true,"
hidden":false,"required":false,"secured":false,"default":"${svc_instance.group}","
customtype":"string","permission":"Read-Only","description":"The current context
group id of the current user that requested\n the provisioning of the template.
This is important because a user may be part of\n more than one group and this
identifies the context of which group the user made the
request.","isSystemTag":true}], "systemTags":true,"creationTime":"2020-07-
13T11:40:11.292Z","lastUpdated":"2020-07-
13T11:40:17.901Z","serviceInstanceCount":2,"id":"5f0c481b415e1137d7efc5fa","action
s":[{"type":"provision","name":"Provision","configuration":
{"onError":"pause"},"input_parameters":[],"flow":{"conditions":[],"templates":
[],"resthooks":[{"resthook":
{"title":"resthook_d4961d","id":"resthook7afd540e","headers":
{"accept":"application/json","Content-
Type":"application/json"},"url":"https://jsonplaceholder.typicode.com/todos/1","me
thod":"GET"}]},"notifications":[],"sequence":
{"0":"resthook7afd540e"},"error_sequence":{"},"abort_sequence":
{},"output_parameters":[],"plans":[{"name":"Standard","description":"To deploy
a Standard plan","actions":[],"plan_parameters":
```



```
[[]], "tagId": "5ecddfe55d6b7600155831de"}, "action_status": {}, "output_parameters":
[], "bind_array": {}}
```

Get the list of deployed instance: GET call

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Where `namespace` is `namespace Id` and `team` is `organization Id`.

Headers required:

- Content-Type: `application/json`
- Authorization: `<bearer token>`

Get details of a Service Instance by using the id

This GET call fetches the details of a specific by its id:

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/<service-instance-
id>?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Headers remain the same as in the terminate service instance call.

Get Input Parameters of a service instance

The API retrieves the inputs parameters of a service instance. It is recommended to use this API before Retry/Resume/Edit service instance so that you can verify the input parameters before Retry/Resume/Edit service instance. This API also resolves the shared parameters and parameters of type connection, if any. Earlier this API (`GET ServiceManifest/5e4f5a51826bc70015dfaeece?service_filter[resolve_params][action_name]=Provision`) was used for the same purpose.

Note: It is recommended to use the `getInputParameters` API instead of `ServiceManifest` API if you want to try Retry or Resume after a service version change.

To use this API, you need a service instance ID. To get the service instance ID:

1. Retrieve all the service instances using the following API [Get the list of deployed instance](#). The response contains an array of service instances
2. Locate the instance by name and get the ID.
3. Use the following curl command to get the input parameters for the service instance:

```
curl --location --request GET \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${ServiceInstan
ceId}/getInputParameters?
cloudOE_spaceGuid=development&ace_orgGuid=all&tenantId=c26b3dd4-4e74-4545-
8ac9-bc30a6167b33' \
--header 'Content-Type: application/json'
```

Response

```
{
  "name": "switch_resthookIns",
  "Status": "Error",
```

```

"owner": "admin",
"active_time": "2020-02-27T05:14:10+00:00",
"templateName": "",
"ordered_time": "2020-02-27T05:13:06.943Z",
"updated_time": "2020-02-27T05:13:06.943Z",
"creationTime": "2020-02-27T05:13:06.943Z",
"instance_id": "d7395b20-591f-11ea-bab8-5f5432f25af5",
"activity_deployment_details": {
  "instance": {
    "vmwarens666346e3": {
      "action": "CREATE",
      "activity_id": "vmwarens666346e3",
      "connection_name": "NSX",
      "created_at": "2020-02-27T05:13:14.914Z",
      "fail_quietly": "false",
      "flow_control_#": "1",
      "id": "5e574feaf63b0a001564e6db",
      "instance_name": "switch_resthookIns-00-inst",
      "long_running_delay": "0",
      "long_running_timeout": "0",
      "provider_engine_version": "0.11.11",
      "resource_id": "serviceblueprint_stack.vmwarens666346e3",
      "sequence_no": "1",
      "stack_data_#": "1",
      "stack_data_latest_state_id": "5e574febf63b0a001564e6dc",
      "stack_error_message": "",
      "stack_id": "5e574feaf63b0a001564e6db",
      "stack_url": "",
      "status": "SUCCESS",
      "status_error_message": "",
      "status_message": "[SUCCESS] ",
      "template_name": "VMware NSX-T Sample to create a logical switch",
      "template_outputs_#": "0",
      "template_params_#": "7",
      "template_params_nsxt_logical_switch_desc": "des",
      "template_params_nsxt_logical_switch_name": "name",
      "template_params_nsxt_logical_switch_repl_mode": "MTEP",
      "template_params_nsxt_logical_switch_state": "UP",
      "template_params_nsxt_tag": "Auto",
      "template_params_nsxt_tag_scope": "Auto-Project",
      "template_params_nsxt_transport_zone": "CAMDC2Zone",
      "template_provider": "VMware NSX-T",
      "template_tags_#": "4",
      "template_tags_request_group": "admin",
      "template_tags_request_user": "admin",
      "template_tags_service_identifier": "5e574fe2dc44840015c8093b",
      "template_tags_service_name": "switch_resthookins",
      "template_type": "TERRAFORM",
      "template_version": "2.3",
      "service_action_type": "provision",
      "service_action_name": "Provision",
      "action_instance_name": "switch_resthookIns",
      "state": "CREATED",
      "flow_control_0.case": "",
      "flow_control_0.case_is_selected": "",
      "flow_control_0.continue_failed": "false",
      "flow_control_0.continue_on_prev_failed": "false",
      "flow_control_0.continue_on_prev_inprogress": "false",
      "flow_control_0.depends_on.#": "1",
      "flow_control_0.depends_on.0": "serviceblueprint_action_start.",
      "flow_control_0.meta.#": "0",
      "flow_control_0.prev_resource_sequence_no": "0",
      "flow_control_0.prev_resource_status": "SUCCESS",
      "flow_control_0.resume": "false",

```

```

        "flow_control_0.resume_version": "0",
        "flow_control_0.version": "0"
    },
    },
    "output_parameters": {}
},
"logs": "serviceblueprint_action_start.start: Refreshing state... (ID:
provision)\nserviceblueprint_stack.vmwarens666346e3: Refreshing state... (ID:
5e574feaf63b0a001564e6db)\nserviceblueprint_http.resthook2ec4d4f7: Refreshing
state... (ID: resthook2ec4d4f7)\nserviceblueprint_http.resthook16dc50cc:
Refreshing state... (ID:
resthook16dc50cc)\nserviceblueprint_http.resthook16dc50cc: Destroying... (ID:
resthook16dc50cc)\nserviceblueprint_http.resthook2ec4d4f7: Modifying... (ID:
resthook2ec4d4f7)\n url: \"https://jsonplaceholder.typicode.com/todos/1\" =>
\"https://jsonplaceholder.Invalid\"\n\nserviceblueprint_http.resthook2ec4d4f7:
Modifications complete after 0s (ID:
resthook2ec4d4f7)\nserviceblueprint_http.resthook16dc50cc: Still
destroying... (ID: resthook16dc50cc, 10s
elapsed)\nserviceblueprint_http.resthook16dc50cc: Destruction complete after
11s\n\n\nApply complete! Resources: 0 added, 1 changed, 1
destroyed.\n\n\nOutputs:\n\n\nresthook2ec4d4f7_message = { \"resthook_88f097\": {
\"status\": \"FAILED\", \"code\": \"502\", \"message\": \"Failure caused
by : <nil>\" }}\n\nresthook2ec4d4f7_status =
FAILED\n\nresthook2ec4d4f7_status_error_message =
<nil>\n\nvmwarens666346e3_message = { \"switch_resthookIns-00-inst\": {
\"status\": \"SUCCESS\", \"action\": \"CREATE\" }}\n\nvmwarens666346e3_status
= SUCCESS\n\nvmwarens666346e3_status_create_error_message =
\n\nvmwarens666346e3_status_error_message =\n\",
    \"serviceId\": \"5e4f59ca826bc70015dfaec8\",
    \"serviceManifestId\": \"5e4f5a51826bc70015dfaece\",
    \"ownership\": {
        \"username\": \"admin\",
        \"displayName\": \"admin\",
        \"orgGuid\": \"all\",
        \"spaceGuid\": \"development\",
        \"tenantId\": \"c26b3dd4-4e74-4545-8ac9-bc30a6167b33\"
    },
    \"namespaceId\": \"development\",
    \"tenantId\": \"c26b3dd4-4e74-4545-8ac9-bc30a6167b33\",
    \"instance_plan\": \"Standard\",
    \"action\": \"Provision\",
    \"id\": \"5e574fe2dc44840015c8093b\",
    \"ServiceID\": \"5e4f59ca826bc70015dfaec8\",
    \"instance_plan_id\": \"5e57566fdc44840015c8094c\",
    \"resourceProperties\": {
        \"templateParameters\": [],
        \"specifications\": [
            {
                \"label\": \"Tansport Zone Name\",
                \"name\": \"nsxt_transport_zone\",
                \"value\": \"CAMDC2Zone\",
                \"secured\": false
            },
            {
                \"label\": \"Rest URL\",
                \"name\": \"url\",
                \"value\": \"https://jsonplaceholder.Invalid\",
                \"secured\": false
            }
        ]
    },
    \"catalogItemId\": \"5e4f59ca826bc70015dfaec8\",
    \"tags_input\": [
        {

```

```

    "value": "admin",
    "label": "request_group",
    "name": "request_group",
    "secured": false
  },
  {
    "value": "admin",
    "label": "request_user",
    "name": "request_user",
    "secured": false
  },
  {
    "value": "switch_resthookins",
    "label": "service_name",
    "name": "service_name",
    "secured": false
  },
  {
    "value": "5e574fe2dc44840015c8093b",
    "label": "service_identifier",
    "name": "service_identifier",
    "secured": false
  }
],
"ace_orgGuid": "all",
"cloudOE_spaceGuid": "development",
"Bearer_Token": "*****",
"bindable": false,
"catalogItemName": "switch_resthook",
"serviceManifest": {
  "version": "2.2.2.2",
  "ServiceEngine_InputFile": {
    "service": {
      "version": "2.2.2.2",
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "switch_resthook",
        "description": "",
        "image": "serviceicon_1.svg",
        "category": "CloudServices",
        "bullets": [],
        "providerDisplayName": "IBM",
        "longDescription": "",
        "documentationUrl": "",
        "supportUrl": "",
        "bindable": false,
        "updatable": "",
        "systemTags": true
      },
      "tags": [
        {
          "name": "request_group",
          "label": "request_group",
          "type": "string",
          "immutable": true,
          "hidden": false,
          "required": false,
          "secured": false,
          "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
          "isSystemTag": true,
          "default": "${svc_instance.group}"
        }
      ]
    }
  }
}

```

```

        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested
the provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end
user at time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is
mapped to the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"upgrade_from": [
    {
        "version": "1.1.1.1"
    }
],
"downgrade_to": [
    {
        "version": "1.1.1.1"
    }
],
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",

```

```

    "input_parameters": [
      {
        "name": "nsxt_transport_zone",
        "label": "Transport Zone Name",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The name of the Transport Zone to be used by
the logical switch.",
        "default": "CAMDC2Zone",
        "editable": true
      },
      {
        "name": "url",
        "label": "Rest URL",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "editable": true,
        "default": "https://jsonplaceholder.typicode.com/todos/1"
      }
    ],
    "flow": {
      "conditions": [],
      "templates": [
        {
          "VMware NSX-T Sample to create a logical switch": {
            "title": "VMware NSX-T Sample to create a logical
switch_b70a6e",
            "template_name": "VMware NSX-T Sample to create a logical
switch",
            "version": "2.3",
            "id": "vmwarens666346e3",
            "template_type": "Terraform",
            "template_content_type": "VMware NSX-T",
            "template_provider": "VMware NSX-T",
            "instance_name": "",
            "cloud_connection_name": "NSX",
            "template_data_objects": {},
            "template_params": {
              "nsxt_tag_scope": "Auto-Project",
              "nsxt_tag": "Auto",
              "nsxt_transport_zone": "CAMDC2Zone",
              "nsxt_logical_switch_name": "name",
              "nsxt_logical_switch_desc": "des",
              "nsxt_logical_switch_state": "UP",
              "nsxt_logical_switch_repl_mode": "MTEP"
            },
            "outputs": [],
            "editable": true,
            "provider_engine_version": "0.11.11"
          }
        }
      ]
    },
    "resthooks": [
      {
        "resthook": {

```

```

        "title": "resthook_88f097",
        "id": "resthook2ec4d4f7",
        "headers": {
            "accept": "application/json",
            "Content-Type": "application/json"
        },
        "editable": true,
        "url": "${input_parameters.url}",
        "method": "GET"
    }
},
"notifications": [],
"sequence": {
    "0": "vmwarens666346e3",
    "1": "resthook2ec4d4f7"
},
"error_sequence": {}
},
"output_parameters": []
}
],
"plans": [
    {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
    }
]
},
"service_id": "5e4f59ca826bc70015dfaec8",
"tenantId": "c26b3dd4-4e74-4545-8ac9-bc30a6167b33",
"namespaceId": "",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that requested the provisioning of the template. This is important because a user may be part of more than one group and this identifies the context of which group the user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,

```

```

        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end
user at time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
    },
    {
        "name": "service_identifer",
        "label": "service_identifer",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifer generated by ICAM which is mapped to
the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
    }
],
"systemTags": true,
"creationTime": "2020-02-21T04:19:29.736Z",
"serviceInstanceCount": 5,
"id": "5e4f5a51826bc70015dfaece",
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [
            {
                "name": "nsxt_transport_zone",
                "label": "Tansport Zone Name",
                "customtype": "string",
                "type": "string",
                "immutable": false,
                "hidden": false,
                "required": false,
                "secured": false,
                "description": "The name of the Transport Zone to be used by the
logical switch.",
                "default": "CAMDC2Zone",
                "editable": true
            },
            {
                "name": "url",

```



```

        "label": "Rest URL",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "editable": true,
        "default": "https://jsonplaceholder.typicode.com/todos/1"
    }
],
"flow": {
    "conditions": [],
    "templates": [
        {
            "VMware NSX-T Sample to create a logical switch": {
                "title": "VMware NSX-T Sample to create a logical
switch_b70a6e",
                "template_name": "VMware NSX-T Sample to create a logical
switch",
                "version": "2.3",
                "id": "vmwarens666346e3",
                "template_type": "Terraform",
                "template_content_type": "VMware NSX-T",
                "template_provider": "VMware NSX-T",
                "instance_name": "",
                "cloud_connection_name": "NSX",
                "template_data_objects": {},
                "template_params": {
                    "nsxt_tag_scope": "Auto-Project",
                    "nsxt_tag": "Auto",
                    "nsxt_transport_zone": "CAMDC2Zone",
                    "nsxt_logical_switch_name": "name",
                    "nsxt_logical_switch_desc": "des",
                    "nsxt_logical_switch_state": "UP",
                    "nsxt_logical_switch_repl_mode": "MTEP"
                },
                "outputs": [],
                "editable": true,
                "provider_engine_version": "0.11.11"
            }
        }
    ],
    "resthooks": [
        {
            "resthook": {
                "title": "resthook_88f097",
                "id": "resthook2ec4d4f7",
                "headers": {
                    "accept": "application/json",
                    "Content-Type": "application/json"
                },
                "editable": true,
                "url": "${input_parameters.url}",
                "method": "GET"
            }
        }
    ],
    "notifications": [],
    "sequence": {
        "0": "vmwarens666346e3",
        "1": "resthook2ec4d4f7"
    }
},

```

```

        "error_sequence": {}
      },
      "output_parameters": []
    }
  ],
  "plans": [
    {
      "name": "Standard",
      "description": "To deploy a Standard plan",
      "actions": [],
      "plan_parameters": []
    }
  ],
  "tagId": "5e4f59ca826bc70015dfaec7"
},
"action_status": {},
"output_parameters": [],
"bind_array": {},
"current_action_instance_id": "5e578abedc44840015c8096c",
"stack_id": [
  "5e574feaf63b0a001564e6db"
],
"last_plan_action_instance_id": "5e578a95dc44840015c8096b",
"last_plan_action_instance_status": "Active",
"InputParameters": {
  "input_parameters": [
    {
      "name": "nsxt_transport_zone",
      "label": "Transport Zone Name",
      "customtype": "string",
      "type": "string",
      "immutable": false,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The name of the Transport Zone to be used by the
logical switch.",
      "default": "CAMDC2Zone",
      "editable": true,
      "value": "CAMDC2Zone"
    },
    {
      "name": "url",
      "label": "Rest URL",
      "customtype": "string",
      "type": "string",
      "immutable": false,
      "hidden": false,
      "required": false,
      "secured": false,
      "isDynamic": false,
      "editable": true,
      "default": "https://jsonplaceholder.typicode.com/todos/1",
      "value": "https://jsonplaceholder.Invalid"
    }
  ],
  "pre_action_input_parameters": [],
  "post_action_input_parameters": []
}
}

```

Edit Service Instance

The `updateServiceInstance` command edits the service instance parameters.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${SERVICE_INSTANCE_ID}/update?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

POST call

Headers:

"Authorization": <bearer token>

"Content-Type": "application/json"

cc_create.json

```
{
  "update_details": {
    "instance_parameters": {
      "in1": "https://jsonplaceholder.typicode.com/todos/22"
    }
  }
}
```

<service_instance_id> - It is the ID of the service instance for which you want to edit.

You must provide all the updated values of service instance parameters to edit this service instance.

Retry Service Instance

The `retryServiceInstance` command attempts to retry all terminate service instances that failed.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${SERVICE_INSTANCE_ID}/retry?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

POST call

Headers:

"Authorization": <bearer token>

"Content-Type": "application/json"

cc_create.json

```
{
  "retry_details": {
    "is_partial": false,
    "instance_parameters": {
      "in1": "https://jsonplaceholder.typicode.com/todos/22"
    }
  }
}
```

<service_instance_id> - It is the ID of the service instance for which you want to retry.

You can run retry with all the updated values for service instance parameters. If the updated service instance parameters are not provided, the values of the previous deployments are used.

If **is_partial** not provided, then by default its value is false. Support is not available for a true value of **is_partial**.

Resume Service Instance

The **resumeServiceInstance** command attempts to resume all terminated service instances that failed.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${SERVICE_INSTANCE_ID}/resume?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

POST call

Headers:

"Authorization": <bearer token>

"Content-Type": "application/json"

Sample cc_create.json:

```
{
  "resume_details": {
    "is_partial": false,
    "instance_parameters": {
      "in1": "https://jsonplaceholder.typicode.com/todos/22"
    }
  }
}
```

<service_instance_id> - It is the ID of the service instance for which you want to resume.

You can resume deployment with all the updated values for service instance parameters. If the updated service instance parameters are not provided, the values of the previous deployments are used.

If **is_partial** not provided, then by default its value is false. Support is not available for a true value of **is_partial**.

Upgrade or downgrade a Service Instance

To upgrade or downgrade a particular service instance to higher or lower service version, the following APIs must be used in sequence:

- [getServiceInstance](#): Returns the service instance details along with the available version change paths (upgrade/downgrade).
- [versionChangeInputParameters](#): Returns the list of input parameters that are needed for service instance version change.
- [submitVersionChange](#): Upgrade/downgrade a version of service instance.

getServiceInstance

This API returns the service instance details along with the available version change paths (upgrade or downgrade).

```
curl --location --request GET \
'https://$TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/5e574fe2dc44840015c8093b?
filter[include]=UpgradePaths&filter[include]=DowngradePaths&cloudOE_spaceGuid=development&ace_orgGuid=all&tenantId=c26b3dd4-4e74-4545-8ac9-bc30a6167b33' \
--header 'Content-Type: application/json'
```

Response

```
{
  {
    "name": "switch_resthookIns",
    "Status": "Error",
    "owner": "admin",
    "active_time": "2020-02-27T05:14:10+00:00",
    "templateName": "",
    "ordered_time": "2020-02-27T05:13:06.943Z",
    "updated_time": "2020-02-27T09:24:47.020Z",
    "creationTime": "2020-02-27T05:13:06.943Z",
    "instance_id": "d7395b20-591f-11ea-bab8-5f5432f25af5",
    "activity_deployment_details": {
      "instance": {
        "vmwarens666346e3": {
          "action": "CREATE",
          "activity_id": "vmwarens666346e3",
          "connection_name": "NSX",
          "created_at": "2020-02-27T05:13:14.914Z",
          "fail_quietly": "false",
          "flow_control_#": "1",
          "id": "5e574feaf63b0a001564e6db",
          "instance_name": "switch_resthookIns-00-inst",
          "long_running_delay": "0",
          "long_running_timeout": "0",
          "provider_engine_version": "0.11.11",
          "resource_id": "serviceblueprint_stack.vmwarens666346e3",
          "sequence_no": "1",
          "stack_data_#": "1",
          "stack_data_latest_state_id": "5e574febf63b0a001564e6dc",
          "stack_error_message": "",
          "stack_id": "5e574feaf63b0a001564e6db",
          "stack_url": "",
          "status": "SUCCESS",
          "status_error_message": "",
          "status_message": "[SUCCESS] ",
          "template_name": "VMware NSX-T Sample to create a logical switch",
          "template_outputs_#": "0",
          "template_params_#": "7",
          "template_params_nsxt_logical_switch_desc": "des",
          "template_params_nsxt_logical_switch_name": "name",
          "template_params_nsxt_logical_switch_repl_mode": "MTEP",
          "template_params_nsxt_logical_switch_state": "UP",
          "template_params_nsxt_tag": "Auto",
          "template_params_nsxt_tag_scope": "Auto-Project",
          "template_params_nsxt_transport_zone": "CAMDC2Zone",
          "template_provider": "VMware NSX-T",
          "template_tags_#": "4",
          "template_tags_request_group": "admin",
          "template_tags_request_user": "admin",
          "template_tags_service_identifier": "5e574fe2dc44840015c8093b",
          "template_tags_service_name": "switch_resthookins",
```

```

        "template_type": "TERRAFORM",
        "template_version": "2.3",
        "service_action_type": "provision",
        "service_action_name": "Provision",
        "action_instance_name": "switch_resthookIns",
        "state": "CREATED",
        "flow_control_0.case": "",
        "flow_control_0.case_is_selected": "",
        "flow_control_0.continue_failed": "false",
        "flow_control_0.continue_on_prev_failed": "false",
        "flow_control_0.continue_on_prev_inprogress": "false",
        "flow_control_0.depends_on.#": "1",
        "flow_control_0.depends_on.0": "serviceblueprint_action_start.",
        "flow_control_0.meta.%": "0",
        "flow_control_0.prev_resource_sequence_no": "0",
        "flow_control_0.prev_resource_status": "SUCCESS",
        "flow_control_0.resume": "false",
        "flow_control_0.resume_version": "0",
        "flow_control_0.version": "0"
    }
},
"output_parameters": {}
},
"logs": "serviceblueprint_action_start.start: Refreshing state... (ID:
provision)\nserviceblueprint_stack.vmwarens666346e3: Refreshing state... (ID:
5e574feaf63b0a001564e6db)\nserviceblueprint_http.resthook2ec4d4f7: Refreshing
state... (ID: resthook2ec4d4f7)\nserviceblueprint_http.resthook16dc50cc:
Refreshing state... (ID:
resthook16dc50cc)\nserviceblueprint_http.resthook16dc50cc: Destroying... (ID:
resthook16dc50cc)\nserviceblueprint_http.resthook2ec4d4f7: Modifying... (ID:
resthook2ec4d4f7)\n url: \"https://jsonplaceholder.typicode.com/todos/1\" =>
\"https://jsonplaceholder.invalid/\"\n\nserviceblueprint_http.resthook2ec4d4f7:
Modifications complete after 0s (ID:
resthook2ec4d4f7)\n\nserviceblueprint_http.resthook16dc50cc: Still destroying...
(ID: resthook16dc50cc, 10s elapsed)\n\nserviceblueprint_http.resthook16dc50cc:
Destruction complete after 11s\n\nApply complete! Resources: 0 added, 1 changed, 1
destroyed.\n\nOutputs:\n\nresthook2ec4d4f7_message = { \"resthook_88f097\": {
\"status\": \"FAILED\", \"code\": \"502\", \"message\": \"Failure caused by :
<nil>\" }}\n\nresthook2ec4d4f7_status =
FAILED\n\nresthook2ec4d4f7_status_error_message = <nil>\n\nvmwarens666346e3_message =
{ \"switch_resthookIns-00-inst\": { \"status\": \"SUCCESS\", \"action\":
\"CREATE\" }}\n\nvmwarens666346e3_status =
SUCCESS\n\nvmwarens666346e3_status_create_error_message =
\n\nvmwarens666346e3_status_error_message =\n\",
    \"serviceId\": \"5e4f59ca826bc70015dfaec8\",
    \"serviceManifestId\": \"5e4f5a51826bc70015dfaee\",
    \"ownership\": {
        \"username\": \"admin\",
        \"displayName\": \"admin\",
        \"orgGuid\": \"all\",
        \"spaceGuid\": \"development\",
        \"tenantId\": \"c26b3dd4-4e74-4545-8ac9-bc30a6167b33\"
    }
},
    \"namespaceId\": \"development\",
    \"tenantId\": \"c26b3dd4-4e74-4545-8ac9-bc30a6167b33\",
    \"instance_plan\": \"Standard\",
    \"action\": \"Provision\",
    \"id\": \"5e574fe2dc44840015c8093b\",
    \"ServiceID\": \"5e4f59ca826bc70015dfaec8\",
    \"instance_plan_id\": \"5e57566fdc44840015c8094c\",
    \"resourceProperties\": {
        \"templateParameters\": [],
        \"specifications\": [
            {

```

```

        "label": "Tansport Zone Name",
        "name": "nsxt_transport_zone",
        "value": "CAMDC2Zone",
        "secured": false
    },
    {
        "label": "Rest URL",
        "name": "url",
        "value": "https://jsonplaceholder.Invalid",
        "secured": false
    }
]
},
"catalogItemId": "5e4f59ca826bc70015dfaec8",
"tags_input": [
    {
        "value": "admin",
        "label": "request_group",
        "name": "request_group",
        "secured": false
    },
    {
        "value": "admin",
        "label": "request_user",
        "name": "request_user",
        "secured": false
    },
    {
        "value": "switch_resthookins",
        "label": "service_name",
        "name": "service_name",
        "secured": false
    },
    {
        "value": "5e574fe2dc44840015c8093b",
        "label": "service_identifier",
        "name": "service_identifier",
        "secured": false
    }
],
"ace_orgGuid": "all",
"cloudOE_spaceGuid": "development",
"Bearer_Token": "*****",
"bindable": false,
"catalogItemName": "switch_resthook",
"serviceManifest": {
    "version": "2.2.2.2",
    "ServiceEngine_InputFile": {
        "service": {
            "version": "2.2.2.2",
            "specVersion": "v3",
            "catalog_metadata": {
                "name": "switch_resthook",
                "description": "",
                "image": "serviceicon_1.svg",
                "category": "CloudServices",
                "bullets": [],
                "providerDisplayName": "IBM",
                "longDescription": "",
                "documentationUrl": "",
                "supportUrl": "",
                "bindable": false,
                "updatable": "",
                "systemTags": true
            }
        }
    }
}

```

```

    },
    "tags": [
      {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "service_name",
        "label": "service_name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user
at time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
      }
    ]
  }

```



```

    ],
    "upgrade_from": [
      {
        "version": "1.1.1.1"
      }
    ],
    "downgrade_to": [
      {
        "version": "1.1.1.1"
      }
    ],
    "actions": [
      {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [
          {
            "name": "nsxt_transport_zone",
            "label": "Transport Zone Name",
            "customtype": "string",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": false,
            "secured": false,
            "description": "The name of the Transport Zone to be used by the
logical switch.",
            "default": "CAMDC2Zone",
            "editable": true
          },
          {
            "name": "url",
            "label": "Rest URL",
            "customtype": "string",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": false,
            "secured": false,
            "isDynamic": false,
            "editable": true,
            "default": "https://jsonplaceholder.typicode.com/todos/1"
          }
        ],
        "flow": {
          "conditions": [],
          "templates": [
            {
              "VMware NSX-T Sample to create a logical switch": {
                "title": "VMware NSX-T Sample to create a logical
switch_b70a6e",
                "template_name": "VMware NSX-T Sample to create a logical
switch",
                "version": "2.3",
                "id": "vmwarens666346e3",
                "template_type": "Terraform",
                "template_content_type": "VMware NSX-T",
                "template_provider": "VMware NSX-T",
                "instance_name": "",
                "cloud_connection_name": "NSX",
                "template_data_objects": {},
                "template_params": {
                  "nsxt_tag_scope": "Auto-Project",

```

```

        "nsxt_tag": "Auto",
        "nsxt_transport_zone": "CAMDC2Zone",
        "nsxt_logical_switch_name": "name",
        "nsxt_logical_switch_desc": "des",
        "nsxt_logical_switch_state": "UP",
        "nsxt_logical_switch_repl_mode": "MTEP"
    },
    "outputs": [],
    "editable": true,
    "provider_engine_version": "0.11.11"
  }
},
"resthooks": [
  {
    "resthook": {
      "title": "resthook_88f097",
      "id": "resthook2ec4d4f7",
      "headers": {
        "accept": "application/json",
        "Content-Type": "application/json"
      },
      "editable": true,
      "url": "${input_parameters.url}",
      "method": "GET"
    }
  },
  {
    "title": "resthook_88f097",
    "id": "resthook2ec4d4f7",
    "headers": {
      "accept": "application/json",
      "Content-Type": "application/json"
    },
    "editable": true,
    "url": "${input_parameters.url}",
    "method": "GET"
  }
],
"notifications": [],
"sequence": {
  "0": "vmwarens666346e3",
  "1": "resthook2ec4d4f7"
},
"error_sequence": {}
},
"output_parameters": []
}
],
"plans": [
  {
    "name": "Standard",
    "description": "To deploy a Standard plan",
    "actions": [],
    "plan_parameters": []
  }
]
}
},
"service_id": "5e4f59ca826bc70015dfaec8",
"tenantId": "c26b3dd4-4e74-4545-8ac9-bc30a6167b33",
"namespaceId": "",
"author": "admin",
"specVersion": "v3",
"status": "Draft",
"tags": [
  {
    "name": "request_group",
    "label": "request_group",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The current context group id of the current user that

```

```

requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
    "isSystemTag": true,
    "default": "${svc_instance.group}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "request_user",
    "label": "request_user",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "default": "${svc_instance.owner}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user at
time of request",
    "isSystemTag": true,
    "default": "${svc_instance.name}",
    "customtype": "string",
    "permission": "Read-Only"
  },
  {
    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is mapped to the
service instance of the template provisioned.",
    "isSystemTag": true,
    "default": "${svc_instance.service_instance_id}",
    "customtype": "string",
    "permission": "Read-Only"
  }
],
"systemTags": true,
"creationTime": "2020-02-21T04:19:29.736Z",
"serviceInstanceCount": 5,
"id": "5e4f5a51826bc70015dfaece",
"actions": [
  {
    "type": "provision",
    "name": "Provision",
    "description": "Default action for deployment of service",
    "input_parameters": [

```

```

    {
      "name": "nsxt_transport_zone",
      "label": "Transport Zone Name",
      "customtype": "string",
      "type": "string",
      "immutable": false,
      "hidden": false,
      "required": false,
      "secured": false,
      "description": "The name of the Transport Zone to be used by the
logical switch.",
      "default": "CAMDC2Zone",
      "editable": true
    },
    {
      "name": "url",
      "label": "Rest URL",
      "customtype": "string",
      "type": "string",
      "immutable": false,
      "hidden": false,
      "required": false,
      "secured": false,
      "isDynamic": false,
      "editable": true,
      "default": "https://jsonplaceholder.typicode.com/todos/1"
    }
  ],
  "flow": {
    "conditions": [],
    "templates": [
      {
        "VMware NSX-T Sample to create a logical switch": {
          "title": "VMware NSX-T Sample to create a logical switch_b70a6e",
          "template_name": "VMware NSX-T Sample to create a logical switch",
          "version": "2.3",
          "id": "vmwarens666346e3",
          "template_type": "Terraform",
          "template_content_type": "VMware NSX-T",
          "template_provider": "VMware NSX-T",
          "instance_name": "",
          "cloud_connection_name": "NSX",
          "template_data_objects": {},
          "template_params": {
            "nsxt_tag_scope": "Auto-Project",
            "nsxt_tag": "Auto",
            "nsxt_transport_zone": "CAMDC2Zone",
            "nsxt_logical_switch_name": "name",
            "nsxt_logical_switch_desc": "des",
            "nsxt_logical_switch_state": "UP",
            "nsxt_logical_switch_repl_mode": "MTEP"
          },
          "outputs": [],
          "editable": true,
          "provider_engine_version": "0.11.11"
        }
      }
    ]
  },
  "resthooks": [
    {
      "resthook": {
        "title": "resthook_88f097",
        "id": "resthook2ec4d4f7",
        "headers": {

```

```

        "accept": "application/json",
        "Content-Type": "application/json"
    },
    "editable": true,
    "url": "${input_parameters.url}",
    "method": "GET"
}
}
],
"notifications": [],
"sequence": {
    "0": "vmwarens666346e3",
    "1": "resthook2ec4d4f7"
},
"error_sequence": {}
},
"output_parameters": []
}
],
"plans": [
{
    "name": "Standard",
    "description": "To deploy a Standard plan",
    "actions": [],
    "plan_parameters": []
}
],
"tagId": "5e4f59ca826bc70015dfaec7"
},
"action_status": {},
"output_parameters": [],
"bind_array": {},
"current_action_instance_id": "5e578abedc44840015c8096c",
"stack_id": [
    "5e574feaf63b0a001564e6db"
],
"last_plan_action_instance_id": "5e578a95dc44840015c8096b",
"last_plan_action_instance_status": "Active",
"actionInstanceDetails": [
{
    "actionInstanceId": "5e574fe3dc44840015c8093c",
    "Status": "Active",
    "processed_time": "2020-02-27T05:13:07.034Z",
    "active_time": "2020-02-27T05:14:10+00:00",
    "ordered_time": 1582780387033,
    "created_time": "2020-02-27T05:14:10+00:00",
    "updated_time": "2020-02-27T05:14:10.982Z",
    "instance_action": "Provision",
    "input_parameters": {}
},
{
    "actionInstanceId": "5e575621dc44840015c80943",
    "Status": "Error",
    "processed_time": "2020-02-27T05:39:45.176Z",
    "ordered_time": 1582781985140,
    "updated_time": "2020-02-27T05:39:45.917Z",
    "instance_action": "plan",
    "versionChangeDetails": {
        "versionChange": true,
        "versionChangeId": "5e50fc1405789a0015a482f5",
        "instance_parameters": {
            "nsxt_transport_zone": "CAMDC2Zone",
            "url": "https://jsonplaceholder.typicode.com/todos/1"
        }
    }
},

```

```

    "pre_action_parameters": {},
    "post_action_parameters": {},
    "fromVersionManifest": {},
    "isRegenerationRequired": true,
    "toVersion": "2.2.2.2"
  },
  "input_parameters": {
    "nsxt_transport_zone": "CAMDC2Zone",
    "url": "https://jsonplaceholder.typicode.com/todos/1"
  }
},
{
  "actionInstanceId": "5e57564fdc44840015c80948",
  "Status": "Error",
  "processed_time": "2020-02-27T05:40:31.198Z",
  "ordered_time": 1582782031165,
  "updated_time": "2020-02-27T05:40:31.649Z",
  "instance_action": "plan",
  "versionChangeDetails": {
    "versionChange": true,
    "versionChangeId": "5e575610dc44840015c8093f",
    "instance_parameters": {
      "nsxt_transport_zone": "CAMDC2Zone",
      "url": "https://jsonplaceholder.typicode.com/todos/1"
    },
    "pre_action_parameters": {},
    "post_action_parameters": {},
    "fromVersionManifest": {},
    "isRegenerationRequired": true,
    "toVersion": "3.3.3.3"
  },
  "input_parameters": {
    "nsxt_transport_zone": "CAMDC2Zone",
    "url": "https://jsonplaceholder.typicode.com/todos/1"
  }
},
{
  "actionInstanceId": "5e5756b8dc44840015c80954",
  "Status": "Active",
  "processed_time": "2020-02-27T05:42:16.162Z",
  "active_time": "2020-02-27T05:42:47+00:00",
  "ordered_time": 1582782136137,
  "created_time": "2020-02-27T05:42:47+00:00",
  "updated_time": "2020-02-27T05:42:47.053Z",
  "instance_action": "plan",
  "versionChangeDetails": {
    "versionChange": true,
    "versionChangeId": "5e57566fdc44840015c8094d",
    "instance_parameters": {
      "nsxt_transport_zone": "CAMDC2Zone",
      "url": "https://jsonplaceholder.typicode.com/todos/1"
    },
    "pre_action_parameters": {},
    "post_action_parameters": {},
    "fromVersionManifest": {},
    "isRegenerationRequired": true,
    "toVersion": "2.2.2.2"
  },
  "input_parameters": {
    "nsxt_transport_zone": "CAMDC2Zone",
    "url": "https://jsonplaceholder.typicode.com/todos/1"
  }
},
{

```

```

    "actionInstanceId": "5e5756eadc44840015c80955",
    "Status": "Active",
    "processed_time": "2020-02-27T05:43:06.719Z",
    "active_time": "2020-02-27T05:43:38+00:00",
    "ordered_time": 1582782186716,
    "created_time": "2020-02-27T05:43:38+00:00",
    "updated_time": "2020-02-27T05:43:38.994Z",
    "instance_action": "update",
    "versionChangeDetails": {
      "serviceInstanceId": "5e574fe2dc44840015c8093b",
      "versionChangeId": "5e57566fdc44840015c8094d",
      "versionChangeType": "UPGRADE",
      "isRegenerationRequired": true,
      "fromVersionId": "5e4f59ca826bc70015dfaec9",
      "fromVersion": "1.1.1.1",
      "fromVersionManifest": {},
      "toVersionId": "5e4f5a51826bc70015dfaece",
      "toVersion": "2.2.2.2",
      "instance_parameters": {
        "nsxt_transport_zone": "CAMDC2Zone",
        "url": "https://jsonplaceholder.typicode.com/todos/1"
      },
      "pre_action_ParmList": [],
      "pre_action_parameters": {},
      "post_action_ParmList": [],
      "post_action_parameters": {}
    },
    "input_parameters": {
      "nsxt_transport_zone": "CAMDC2Zone",
      "url": "https://jsonplaceholder.typicode.com/todos/1"
    }
  },
  {
    "actionInstanceId": "5e57891bdc44840015c80969",
    "Status": "Active",
    "processed_time": "2020-02-27T09:17:15.511Z",
    "active_time": "2020-02-27T09:17:47+00:00",
    "ordered_time": 1582795035480,
    "created_time": "2020-02-27T09:17:47+00:00",
    "updated_time": "2020-02-27T09:17:47.695Z",
    "instance_action": "plan",
    "versionChangeDetails": {
      "versionChange": true,
      "versionChangeId": "5e575d04dc44840015c80963",
      "instance_parameters": {
        "url": "https://jsonplaceholder.typicode.com/todos/1",
        "url2": "https://jsonplaceholder.typicode.com/todos/2"
      },
      "pre_action_parameters": {},
      "post_action_parameters": {},
      "fromVersionManifest": {},
      "isRegenerationRequired": true,
      "toVersion": "3.3.3.3"
    },
    "input_parameters": {
      "url": "https://jsonplaceholder.typicode.com/todos/1",
      "url2": "https://jsonplaceholder.typicode.com/todos/2"
    }
  },
  {
    "actionInstanceId": "5e578990dc44840015c8096a",
    "Status": "Active",
    "processed_time": "2020-02-27T09:19:12.978Z",
    "active_time": "2020-02-27T09:19:46+00:00",

```

```

"ordered_time": 1582795152977,
"created_time": "2020-02-27T09:19:46+00:00",
"updated_time": "2020-02-27T09:19:46.727Z",
"instance_action": "update",
"versionChangeDetails": {
  "serviceInstanceId": "5e574fe2dc44840015c8093b",
  "versionChangeId": "5e575d04dc44840015c80963",
  "versionChangeType": "UPGRADE",
  "isRegenerationRequired": true,
  "fromVersionId": "5e4f5a51826bc70015dfaece",
  "fromVersion": "2.2.2.2",
  "fromVersionManifest": {},
  "toVersionId": "5e575610dc44840015c8093d",
  "toVersion": "3.3.3.3",
  "instance_parameters": {
    "url1": "https://jsonplaceholder.typicode.com/todos/1",
    "url2": "https://jsonplaceholder.typicode.com/todos/2"
  },
  "pre_action_ParmList": [],
  "pre_action_parameters": {},
  "post_action_ParmList": [],
  "post_action_parameters": {}
},
"input_parameters": {
  "url1": "https://jsonplaceholder.typicode.com/todos/1",
  "url2": "https://jsonplaceholder.typicode.com/todos/2"
}
},
{
  "actionInstanceId": "5e578a95dc44840015c8096b",
  "Status": "Active",
  "processed_time": "2020-02-27T09:23:33.091Z",
  "active_time": "2020-02-27T09:24:05+00:00",
  "ordered_time": 1582795413060,
  "created_time": "2020-02-27T09:24:05+00:00",
  "updated_time": "2020-02-27T09:24:05.271Z",
  "instance_action": "plan",
  "versionChangeDetails": {
    "versionChange": true,
    "versionChangeId": "5e575d04dc44840015c80964",
    "instance_parameters": {
      "nsxt_transport_zone": "CAMDC2Zone",
      "url": "https://jsonplaceholder.Invalid"
    },
    "pre_action_parameters": {},
    "post_action_parameters": {},
    "fromVersionManifest": {},
    "isRegenerationRequired": true,
    "toVersion": "2.2.2.2"
  },
  "input_parameters": {
    "nsxt_transport_zone": "CAMDC2Zone",
    "url": "https://jsonplaceholder.Invalid"
  }
},
{
  "actionInstanceId": "5e578abedc44840015c8096c",
  "Status": "Error",
  "processed_time": "2020-02-27T09:24:14.187Z",
  "ordered_time": 1582795454186,
  "updated_time": "2020-02-27T09:24:47.020Z",
  "instance_action": "update",
  "versionChangeDetails": {
    "serviceInstanceId": "5e574fe2dc44840015c8093b",

```



```

    "versionChangeId": "5e575d04dc44840015c80964",
    "versionChangeType": "DOWNGRADE",
    "isRegenerationRequired": true,
    "fromVersionId": "5e575610dc44840015c8093d",
    "fromVersion": "3.3.3.3",
    "fromVersionManifest": {},
    "toVersionId": "5e4f5a51826bc70015dfaace",
    "toVersion": "2.2.2.2",
    "instance_parameters": {
      "nsxt_transport_zone": "CAMDC2Zone",
      "url": "https://jsonplaceholder.Invalid"
    },
    "pre_action_ParmList": [],
    "pre_action_parameters": {},
    "post_action_ParmList": [],
    "post_action_parameters": {}
  },
  "input_parameters": {
    "nsxt_transport_zone": "CAMDC2Zone",
    "url": "https://jsonplaceholder.Invalid"
  }
},
],
"processed_time": "2020-02-27T09:24:14.187Z",
"output": {},
"UpgradePaths": [
  {
    "currentVersionId": "5e575610dc44840015c8093d",
    "fromVersionId": "5e4f5a51826bc70015dfaace",
    "toVersionId": "5e575610dc44840015c8093d",
    "type": "UPGRADE",
    "tenantId": "c26b3dd4-4e74-4545-8ac9-bc30a6167b33",
    "id": "5e575d04dc44840015c80963"
  }
],
"DowngradePaths": [
  {
    "currentVersionId": "5e4f5a51826bc70015dfaace",
    "fromVersionId": "5e4f5a51826bc70015dfaace",
    "toVersionId": "5e4f59ca826bc70015dfaec9",
    "type": "DOWNGRADE",
    "tenantId": "c26b3dd4-4e74-4545-8ac9-bc30a6167b33",
    "id": "5e57566fdc44840015c8094e"
  }
]
}

```

Note: id present inside the UpgradePaths or DowngradePaths should be used as **versionChangeId** in payloads of the **versionChangeInputParameters** and **submitVersionChange** APIs.

versionChangeInputParameters

This API returns the list of input parameters that are needed for service instance version change.

```

curl --location --request POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/5e574fe2dc44840015c8093b/versionChangeInputParameters?cloudOE_spaceGuid=development&ace_orgGuid=all&tenantId=c26b3dd4-4e74-4545-8ac9-bc30a6167b33' \
--header 'Content-Type: application/json' \
--data-raw '{
  "versionChangeId": "5e575d04dc44840015c80963"
}'

```

versionChangeId is the ID that is present inside the **UpgradePaths** or **DowngradePaths**, which is returned with the **getServiceInstance** API.

Request Body

```
{
  "versionChangeId": "5e575c44dc44840015c80959"
}
```

Response

```
{
  "VersionChangeInputParameters": {
    "versionChangeId": "5e575d04dc44840015c80963",
    "toVersion": "3.3.3.3",
    "type": "UPGRADE",
    "input_parameters": [
      {
        "name": "url",
        "label": "Rest URL",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "editable": true,
        "default": "https://jsonplaceholder.Invalid"
      },
      {
        "name": "url2",
        "label": "URL 2",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "editable": true,
        "default": "https://jsonplaceholder.typicode.com/todos/2"
      }
    ],
    "pre_action_input_parameters": [],
    "post_action_input_parameters": [],
    "new_parameters": [
      "url2"
    ],
    "used_old_parameters": [
      "url"
    ],
    "ignored_old_parameters": [
      "nsxt_transport_zone"
    ]
  }
}
```

Note:

- **input_parameters** - A list of input parameters needed for service instance version change.
- **new_parameters** - A list of new input parameters present in the new version.

- **used_old_parameters**- A list of input parameters present from the old version which are used in new version.
- **ignored_old_parameters** - A list of input parameters present from the old version which are not used in new version.
- **pre_action_input_parameters** and **post_action_input_parameters** are not supported in this release.

submitVersionChange

This API upgrades/downgrades a version of the service instance.

```
curl --location --request POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/5e574fe2dc44840015c8093b/submitVersionChange?cloudOE_spaceGuid=development&ace_orgGuid=all&tenantId=c26b3dd4-4e74-4545-8ac9-bc30a6167b33' \
--header 'Content-Type: application/json' \
--data-raw '{
  "versionChangeId": "5e575d04dc44840015c80963",
  "instance_parameters": {
    "url": "https://jsonplaceholder.typicode.com/todos/1",
    "url2": "https://jsonplaceholder.typicode.com/todos/2"
  },
  "pre_action_parameters": {},
  "post_action_parameters": {},
  "plan": ""
}
```

Request Body

```
{
  "versionChangeId": "5e575d04dc44840015c80963",
  "instance_parameters": {
    "url": "https://jsonplaceholder.typicode.com/todos/1",
    "url2": "https://jsonplaceholder.typicode.com/todos/2"
  },
  "pre_action_parameters": {},
  "post_action_parameters": {},
  "plan": ""
}
```

Bind a service instance

The API call creates a bind instance.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${SERVICE_INSTANCE_ID}/createBind?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}'
```

POST call

Headers:

"Authorization": "bearer <token>"

"Content-Type": "application/json"

cc_create.json

```
{
  "bindDetails": {
    "instance_parameters": {
      "resturl2": "https://jsonplaceholder.typicode.com/todos/33"
    }
  }
}
```

- **<service_instance_id>** - It is the ID of the service instance for which you want to create a bind instance.
- **<instance_parameters>** - Optionally, you can run bind with updated values for service instance parameters. If not provided, then earlier values are used.

Unbind Instance

The API call unbinds or deletes a bind instance.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${SERVICE_INSTANCE_ID}/terminateBind?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

BODY:

cc_create.json

```
"name": "<name_of_bind_instance>"}
```

<name_of_bind_instance> - It is the name of the bind instance that you want to unbind.

Get custom actions

List the custom actions of a service instance.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/<serviceInstanceId>/getCustomActions?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

This GET call fetches the list of custom actions of service instance.

Headers:

```
"Authorization": "bearer <token>"
"Content-Type": "application/json"
```

Response

```
{
  "customActions": [
    {
      "type": "custom",
      "name": "custom1",
      "description": "customaction",
      "pre_or_post": false,
      "input_parameters": [
        {
          "name": "resthook_url",
```

```

        "label": "resthook_url",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false
    }
],
"flow": {
    "conditions": [],
    "templates": [],
    "resthooks": [
        {
            "resthook": {
                "title": "resthook_f428a5",
                "id": "resthook56a3f1fc",
                "headers": {
                    "accept": "application/json",
                    "Content-Type": "application/json"
                },
                "editable": true,
                "url": "${input_parameters.resthook_url}",
                "method": "GET"
            }
        }
    ],
    "notifications": [],
    "sequence": {
        "0": "resthook56a3f1fc"
    },
    "error_sequence": {}
},
"output_parameters": []
}
]
}

```

Invoke custom action

To execute a custom action on a service instance.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @custom_action.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/<serviceInstanceId>/
invokeCustomAction?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"

```

POST call

Headers:

```

"Authorization": "bearer <token>"
"Content-Type": "application/json"

```

custom_action.json

```

{
    "customDetails": {
        "serviceInstanceId": "<serviceInstanceId>"
    }
}

```

```

    "name": "actionInstanceName",
    "instance_parameters": {
      "resthook_url": "https://jsonplaceholder.typicode.com/todos/33"
    },
    "currentCustomAction": "<actionName>"
  }
}

```

instance_parameters - Optionally, you can run custom action with updated values for service instance parameters. The default values are used, if parameters are not provided.

Terminate Service Instance

```

curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/${ServiceInstanceId}/terminate?ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @service__instance_terminate.json

```

Content of service__instance_terminate.json:

```
{}
```

Response

```

{"name": "upgarde_downgrade_service_instance56857d10-0220-11eb-859b-adea70bdca6d", "Status": "Terminating", "id": "5f72d9ac2401e29758629ce9"}

```

Cleanup Service Instance

It deletes all service instances based on the value of **force** flag. If the **force = false**, then the instances in active state are not deleted. The instances in error state are deleted irrespective of the **force** flag value.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/{{SERVICE_INSTANCE_ID}}/cleanup?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}'

```

POST call

Headers:

```

"Authorization": <bearer token>
"Content-Type": "application/json"

```

```

Body: {
  "force": <true/false>
}

```

Cleanup Service Instance in inprogress state

If a service instance is stuck in IN_PROGRESS state for a long time without any backend activity, then use this API to delete the service instance entry from the database.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceInstances/{{SERVICE_INSTANCE_ID}}/cleanup?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Sample Request BODY:

```
{
  "serviceInstanceId": "5cf0c20cbdf17d001786d1a9",
  "force": true
}
```

The **force = true** deletes the service instance entry from the Managed services database forcefully. However, the activities that are running does not get cleaned up and they throw the following error exception:

```
{
  "error": {
    "statusCode": "419",
    "name": "Error",
    "message": "ResourceLockError: Failed to obtain lock for resource stack_e19dd5d0-85f5-11e9-8609-cf2b57a8a189.; caused by Error: unable to obtain lock"
  }
}
```

The **ResourceLockError** is related to Mongo DB lock. Whenever any deployment starts, Mongo DB lock associated with the stack_id is placed and it remains locked for 1 hour. If service instance is stuck and the cleanup API gives ResourceLockError, try the same API after 1 hour, that is, after the lock expires. It is a precautionary step to avoid cleaning up of service instances that are in running state at the backend, which might cause inconsistency. Also, there can be side effects if an instance having dangled resources gets cleaned up.

Standalone action APIs

Create Standalone Action

Creates a stand-alone action independent of any service.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" -H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/Actions/createAction?tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid={NAMESPACE}"
```

The **cloudOE_spaceGuid** contains the value of the namespace. Note that the **NAMESPACE** must be the desired namespace to deploy to.

POST method

Example BODY:

```
cc_create.json

{
  "type": "provision",
  "name": "newAct44f",
}
```

```

"description": "Default action for deployment of service",
"input_parameters": [],
"flow": {
  "conditions": [],
  "templates": [
    {
      "Template With Output Parameters": {
        "title": "Template With Output Parameters_343311",
        "template_name": "Template With Output Parameters ce0c21",
        "version": "master",
        "id": "template46b48313",
        "template_type": "Terraform",
        "template_content_type": "VMware vSphere",
        "template_provider": "VMware vSphere",
        "instance_name": "",
        "cloud_connection_name": "Global-VV1",
        "template_data_objects": {},
        "template_params": {
          "sleep": "10000",
          "cam-ip": "9.5.29.229",
          "data-server": "9.5.28.159",
          "data-as-query-string": "id=1&name=test",
          "region": "us-east-1"
        },
        "outputs": [
          "resource-id",
          "selected-region",
          "data"
        ]
      }
    }
  ],
  "resthooks": [],
  "notifications": [],
  "sequence": {
    "0": "template46b48313"
  }
},
"output_parameters": []
}

```

After you create an action, an action ID is generated automatically:

Example response:

```

{
  "ServiceEngine_InputFile": {
    "service": {
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "dsfdfs",
        "description": "",
        "image": "serviceicon_1.svg",
        "category": "CloudServices",
        "bullets": [],
        "providerDisplayName": "IBM",
        "longDescription": "",
        "documentationUrl": "",
        "supportUrl": "",
        "bindable": false,
        "updatable": "",
        "systemTags": true,
        "author": "admin"
      }
    }
  }
}

```



```

    },
    "tags": [
      {
        "name": "Request_Group",
        "label": "Request_Group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current user
that requested the provisioning of the template. This is important because a user
may be part of more than one group and this identifies the context of which group
the user made the request.",
        "isSystemTag": true,
        "default": "${svc_instance.group}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "Request_User",
        "label": "Request_User",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested the
provisioning of the template.",
        "isSystemTag": true,
        "default": "${svc_instance.owner}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "Service_Name",
        "label": "Service_Name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end
user at time of request",
        "isSystemTag": true,
        "default": "${svc_instance.name}",
        "customtype": "string",
        "permission": "Read-Only"
      },
      {
        "name": "Service_Identifier",
        "label": "Service_Identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped
to the service instance of the template provisioned.",
        "isSystemTag": true,
        "default": "${svc_instance.service_instance_id}",
        "customtype": "string",
        "permission": "Read-Only"
      }
    ]
  }

```

```

    ],
    "actions": [
      {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [],
        "flow": {
          "conditions": [],
          "templates": [
            {
              "Amazon EC2 Virtual Server with SSH key": {
                "title": "Amazon EC2 Virtual Server with SSH
key_25e3a0",
                "template_name": "Amazon EC2 Virtual Server with SSH
key",
                "version": "2.3",
                "id": "amazoneceeb670ee",
                "template_type": "Terraform",
                "template_content_type": "Amazon EC2",
                "template_provider": "Amazon EC2",
                "instance_name": "",
                "cloud_connection_name": "",
                "template_data_objects": {},
                "template_params": {
                  "": "us-east-1",
                  "aws_image": "ubuntu/images/hvm-ssd/ubuntu-trusty-
14.04-amd64-server-*",
                  "aws_image_size": "t2.small",
                  "aws_ami_owner_id": "099720109477",
                  "vpc_name_tag": "",
                  "subnet_name": "",
                  "public_ssh_key_name": "",
                  "public_ssh_key": ""
                },
                "outputs": [],
                "warning": true,
                "current_action_key": 0
              }
            }
          ],
          "resthooks": [],
          "notifications": [],
          "sequence": {
            "0": "amazoneceeb670ee"
          },
          "error_sequence": {}
        },
        "output_parameters": []
      }
    ],
    "plans": [
      {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
      }
    ],
    "tagId": "5c36d79c3abef7001aead045"
  }
},
"service_id": "5cf5fd6e30483a00144c919f",
"namespaceId": "",

```

```

    "id": "5cf5fd6e30483a00144c919e",
    "warning": [
      "Template doesnt exists or no access: response.statusCode: 400",
      " Template amazoneceeb670ee does not have cloud connection name.",
      "Templates- 'template_params' of type json object should have non-empty keys",
      "template.id : amazoneceeb670ee"
    ]
  }
}

```

For complete format of the action, see [action section in the Editing a service file](#) section.

Invoke Standalone Action

Use the action id generated in the **Create Action** API call to invoke an action.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create.json
"https://${TSA_HOSTNAME}/cam/composer/api/v1/ActionInstances/invokeAction?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid={NAMESPACE}"

```

The `cloudOE_spaceGuid` contains the value of the namespace. Note that the **NAMESPACE** must be the desired namespace to deploy to.

POST method

Example BODY:

```

"action_data": {
  "action_id": "5caf3b64b52b5a00182116af",
  "name": "inst2 "
}

```

Example response:

```

{
  "actionInstance": {
    "id": "5cd01a095253de001918ce2c"
  }
}

```

It returns Action instance ID.

ICON APIs

Upload service icons

This API uploads new icons. Select the file to upload.

POST method

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-F "icon=@PATH_TO_FILE" "https://${TSA_HOSTNAME}/cam/composer/api/v1/icons/upload?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"

```

It returns the name and id of the image.

BODY example:

```
{
  "res": {
    "name": "serviceicon_1.svg",
    "id": "5czdb1bc1b47001cbc556"
  }
}
```

Note:

- The size of the icon file must be less than one MB.
- The name of the icon must be unique.
- The supported content types are as follows:
CONTENT_TYPE = ['image/jpeg', 'image/gif', 'image/png', 'application/xml'];
- Upload images with the key name as 'icon'.

Delete service icons

This API deletes service icons.

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/icons/${ICON_ID}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Retrieve all service icons

This API returns metadata of all available icons.

GET method.

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/icons?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

BODY example:

```
[
  {
    "ContentType": "application/xml",
    "name": "serviceicon_1.svg",
    "id": "5czdb1bc1b47001cbc556"
  },
  {
    "ContentType": "application/xml",
    "name": "serviceicon_2.svg",
    "id": "5czdb1bc1b47001cbc557"
  },
  {
    "ContentType": "application/xml",
    "name": "serviceicon_3.svg",
    "id": "SCZ5czdb1bc1b47001cbc559"
  }
]
```

You can also set the filter to retrieve the metadata of a service icon based on its name, ID, or type.

```
GET /cam/composer/api/v1/icons/?tenantId=62c6ce6e-1f62-4fe1-b176-
259037a7bff2&ace_orgGuid=all&filter[where][name]=serviceicon_1.svg
```

BODY example:

```
{
  {
    "ContentType": "application/xml",
    "name": "serviceicon_1.svg",
    "id": "5czdb1bc1b47001cbc556"
  }
}
```

Download service icon

This API call downloads the specified icon.

GET method

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
"https://${TSA_HOSTNAME}/cam/composer/api/v1/icons/name/${ICON_NAME}?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Custom action APIs

This section includes all the custom action APIs with examples.

- [Create a custom action category](#)
- [Create a custom action](#)
- [Lock a custom action](#)
- [Unlock a custom action](#)
- [Get the list of custom actions](#)
- [Assign a custom action to service version \(service manifest\)](#)
 - [Additional information for the custom action to a service version \(service manifest\) API](#)

Custom action APIs

Create a custom action category

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/customActionCategories?
ace_orgGuid=${MCM_TEAM}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d {"Name": "demo-category"}
```

Response

```
{
  "Name": "demo-category",
  "id": "621cc8e6218004001ac4bc00",
  "tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380"
}
```

Create a custom action

```
curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/CustomAction/createCustomAction?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @custom_action_create.json
```

custom_action_create.json`

```
{
  "action": {
    "type": "custom",
    "name": "ind-action",
    "category": "demo-category",
    "description": " Independent custom action",
    "input_parameters": [
      {
        "name": "inputurl",
        "label": "inputpurl",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "default": "https://jsonplaceholder.typicode.com/todos/1"
      }
    ],
    "flow": {
      "conditions": [],
      "templates": [],
      "resthooks": [
        {
          "resthook": {
            "title": "resthook_354d4a",
            "id": "resthook98f44a65",
            "headers": {
              "accept": "application/json",
              "Content-Type": "application/json"
            },
            "editable": true,
            "url": "${input_parameters.inputurl}",
            "method": "GET"
          }
        }
      ],
      "notifications": [],
      "sequence": {
        "0": "resthook98f44a65"
      },
      "error_sequence": {}
    },
    "output_parameters": [
      {
        "name": "resthookOutput",
        "label": "resthookoutput",
        "customtype": "string",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
```

```

        "secured": false,
        "description": "",
        "default": "${resthooks.resthook98f44a65.output.body}"
    }
  ]
}
}

```

Response

```

{
  "data": {
    "name": "ind-action",
    "ServiceEngine_InputFile": {
      "action": {
        "type": "custom",
        "name": "ind-action",
        "category": "demo-category",
        "description": " Independent custom action",
        "input_parameters": [
          {
            "name": "inputurl",
            "label": "inputpurl",
            "customtype": "string",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": false,
            "secured": false,
            "isDynamic": false,
            "default": "https://jsonplaceholder.typicode.com/todos/1"
          }
        ],
        "flow": {
          "conditions": [],
          "templates": [],
          "resthooks": [
            {
              "resthook": {
                "title": "resthook_354d4a",
                "id": "resthook98f44a65",
                "headers": {
                  "accept": "application/json",
                  "Content-Type": "application/json"
                },
                "editable": true,
                "url": "${input_parameters.inputurl}",
                "method": "GET"
              }
            }
          ],
          "notifications": [],
          "sequence": {
            "0": "resthook98f44a65"
          },
          "error_sequence": {}
        },
        "output_parameters": [
          {
            "name": "resthookOutput",
            "label": "resthookoutput",
            "customtype": "string",
            "type": "string",
            "immutable": true,

```

```

        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "default": "${resthooks.resthook98f44a65.output.body}"
    }
}
    ],
    },
    "categoryId": "621cc8e6218004001ac4bc00",
    "namespaceId": "cp4aiops",
    "tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380",
    "creationTime": "2022-02-28T13:13:30.918Z",
    "lastUpdated": "2022-02-28T13:13:30.672Z",
    "status": "Unlocked",
    "id": "621cca7a218004001ac4bc03",
    "warning": []
}
}
}

```

Lock a custom action

```

curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/CustomAction/<customid>/lock?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d {}

```

Response:

```

{
  "data": {
    "name": "ind-action",
    "ServiceEngine_InputFile": {
      "action": {
        "type": "custom",
        "name": "ind-action",
        "category": "demo-category",
        "description": "Independent custom action",
        "input_parameters": [
          {
            "name": "inputurl",
            "label": "inputpurl",
            "customtype": "string",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": false,
            "secured": false,
            "isDynamic": false,
            "default": "https://jsonplaceholder.typicode.com/todos/1"
          }
        ],
        "flow": {
          "conditions": [],
          "templates": [],
          "resthooks": [
            {
              "resthook": {
                "title": "resthook_354d4a",

```



```

        "id": "resthook98f44a65",
        "headers": {
            "accept": "application/json",
            "Content-Type": "application/json"
        },
        "editable": true,
        "url": "${input_parameters.inputurl}",
        "method": "GET"
    }
}
],
"notifications": [],
"sequence": {
    "0": "resthook98f44a65"
},
"error_sequence": {}
},
"output_parameters": [
    {
        "name": "resthookOutput",
        "label": "resthookoutput",
        "customtype": "string",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "default": "${resthooks.resthook98f44a65.output.body}"
    }
]
}
},
"categoryId": "621cc8e6218004001ac4bc00",
"namespaceId": "cp4aiops",
"tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380",
"creationTime": "2022-02-28T13:13:30.918Z",
"lastUpdated": "2022-02-28T13:20:01.513Z",
"status": "Locked",
"id": "621cca7a218004001ac4bc03"
}
}

```

Get the list of custom actions

```

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}"
https://${TSA_HOSTNAME}/cam/composer/api/v1/CustomAction?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"

```

Response

```

[
    {
        "name": "ind-action",
        "ServiceEngine_InputFile": {
            "action": {
                "type": "custom",
                "name": "ind-action",
                "category": "demo-category",
                "description": " Independent custom action",
                "input_parameters": [
                    {

```

```

        "name": "inputurl",
        "label": "inputpurl",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "isDynamic": false,
        "default": "https://jsonplaceholder.typicode.com/todos/1"
    },
    ],
    "flow": {
        "conditions": [],
        "templates": [],
        "resthooks": [
            {
                "resthook": {
                    "title": "resthook_354d4a",
                    "id": "resthook98f44a65",
                    "headers": {
                        "accept": "application/json",
                        "Content-Type": "application/json"
                    },
                    "editable": true,
                    "url": "${input_parameters.inputurl}",
                    "method": "GET"
                }
            }
        ],
        "notifications": [],
        "sequence": {
            "0": "resthook98f44a65"
        },
        "error_sequence": {}
    },
    "output_parameters": [
        {
            "name": "resthookOutput",
            "label": "resthookoutput",
            "customtype": "string",
            "type": "string",
            "immutable": true,
            "hidden": false,
            "required": false,
            "secured": false,
            "description": "",
            "default": "${resthooks.resthook98f44a65.output.body}"
        }
    ]
},
{
    "categoryId": "621cc8e6218004001ac4bc00",
    "namespaceId": "cp4aiops",
    "tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380",
    "creationTime": "2022-02-28T13:13:30.918Z",
    "lastUpdated": "2022-02-28T13:22:09.539Z",
    "status": "Unlocked",
    "id": "621cca7a218004001ac4bc03",
    "serviceVersionCount": 0,
    "serviceVersionNames": []
}
]

```

Assign a custom action to service version (service manifest)

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/ServiceManifest/<servicemanifestid>/assignCustomActions?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @assign_custom_actions.json
```

assign_custom_actions.json:

```
{
  "custom_action_details": [
    {
      "customActionName": "ind-action",
      "customActionId": "<custom action id>",
      "customInputParams": {
        "inputurl": "${svc_instance.output_parameters.todosUrl3}"
      }
    }
  ]
}
```

Response:

```
{
  "version": "1.0.0.0",
  "ServiceEngine_InputFile": {
    "service": {
      "version": "1.0.0.0",
      "specVersion": "v3",
      "catalog_metadata": {
        "name": "assignAService",
        "description": "",
        "image": "serviceicon_1.svg",
        "category": "dinesh",
        "bullets": [],
        "providerDisplayName": "IBM",
        "longDescription": "",
        "documentationUrl": "",
        "supportUrl": "",
        "bindable": false,
        "updatable": "",
        "systemTags": true
      },
      "tags": [
        {
          "name": "environment",
          "label": "Environment",
          "type": "string",
          "immutable": false,
          "hidden": false,
          "required": true,
          "secured": false,
          "description": "Unique identifier added to the provisioned VMs.",
          "isSystemTag": true,
          "customtype": "string",
          "permission": "Read-Write",

```

```

        "options": [
            {
                "label": "Development",
                "value": "dev",
                "default": true
            },
            {
                "label": "Quality Assurance",
                "value": "qa",
                "default": false
            },
            {
                "label": "Production",
                "value": "prod",
                "default": false
            },
            {
                "label": "Test",
                "value": "test",
                "default": false
            }
        ]
    },
    {
        "name": "Name",
        "label": "Name",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the
end user at time of request",
        "isSystemTag": true,
        "customtype": "string",
        "permission": "Read-Only",
        "default": "${svc_instance.name}"
    },
    {
        "name": "request_group",
        "label": "request_group",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The current context group id of the current
user that requested the provisioning of the template. This is important because a
user may be part of more than one group and this identifies the context of which
group the user made the request.",
        "isSystemTag": true,
        "customtype": "string",
        "permission": "Read-Only",
        "default": "${svc_instance.group}"
    },
    {
        "name": "request_user",
        "label": "request_user",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "The user id of the current user that requested

```

```

the provisioning of the template.",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.owner}"
  },
  {
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the
end user at time of request",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.name}"
  },
  {
    "name": "service_identifier",
    "label": "service_identifier",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Unique identifier generated by ICAM which is
mapped to the service instance of the template provisioned.",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.service_instance_id}"
  }
],
"upgrade_from": [],
"downgrade_to": [],
"actions": [
  {
    "type": "provision",
    "name": "Provision",
    "description": "Default action for deployment of service",
    "input_parameters": [],
    "flow": {
      "conditions": [],
      "templates": [],
      "resthooks": [
        {
          "resthook": {
            "title": "resthook_cfb688",
            "id": "resthook00fle7c5",
            "headers": {
              "accept": "application/json",
              "Content-Type": "application/json"
            },
            "editable": true,
            "url":
"https://jsonplaceholder.typicode.com/todos/2",
            "method": "GET"
          }
        }
      ]
    }
  }
]

```

```

        "notifications": [],
        "sequence": {
            "0": "resthook00f1e7c5"
        },
        "error_sequence": {}
    },
    "output_parameters": [
        {
            "name": "todosUrl3",
            "label": "todosUrl3",
            "customtype": "string",
            "type": "string",
            "immutable": true,
            "hidden": false,
            "required": false,
            "secured": false,
            "default":
"https://jsonplaceholder.typicode.com/todos/3"
        }
    ]
},
    "plans": [
        {
            "name": "Standard",
            "description": "To deploy a Standard plan",
            "actions": [],
            "plan_parameters": []
        }
    ]
},
    "service_id": "621ccec6218004001ac4bc04",
    "tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380",
    "namespaceId": "",
    "author": "admin",
    "specVersion": "v3",
    "status": "Published",
    "publishedTime": "2022-02-28T13:33:08.798Z",
    "tags": [
        {
            "name": "environment",
            "label": "Environment",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": true,
            "secured": false,
            "description": "Unique identifier added to the provisioned VMs.",
            "isSystemTag": true,
            "customtype": "string",
            "permission": "Read-Write",
            "options": [
                {
                    "label": "Development",
                    "value": "dev",
                    "default": true
                },
                {
                    "label": "Quality Assurance",
                    "value": "qa",
                    "default": false
                }
            ]
        }
    ]
}

```

```

        "label": "Production",
        "value": "prod",
        "default": false
    },
    {
        "label": "Test",
        "value": "test",
        "default": false
    }
]
},
{
    "name": "Name",
    "label": "Name",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "Name provided for the service instance by the end user
at time of request",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.name}"
},
{
    "name": "request_group",
    "label": "request_group",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The current context group id of the current user that
requested the provisioning of the template. This is important because a user may
be part of more than one group and this identifies the context of which group the
user made the request.",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.group}"
},
{
    "name": "request_user",
    "label": "request_user",
    "type": "string",
    "immutable": true,
    "hidden": false,
    "required": false,
    "secured": false,
    "description": "The user id of the current user that requested the
provisioning of the template.",
    "isSystemTag": true,
    "customtype": "string",
    "permission": "Read-Only",
    "default": "${svc_instance.owner}"
},
{
    "name": "service_name",
    "label": "service_name",
    "type": "string",
    "immutable": true,
    "hidden": false,

```

```

        "required": false,
        "secured": false,
        "description": "Name provided for the service instance by the end user
at time of request",
        "isSystemTag": true,
        "customtype": "string",
        "permission": "Read-Only",
        "default": "${svc_instance.name}"
    },
    {
        "name": "service_identifier",
        "label": "service_identifier",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "Unique identifier generated by ICAM which is mapped to
the service instance of the template provisioned.",
        "isSystemTag": true,
        "customtype": "string",
        "permission": "Read-Only",
        "default": "${svc_instance.service_instance_id}"
    }
],
"systemTags": true,
"creationTime": "2022-02-28T13:31:50.023Z",
"lastUpdated": "2022-02-28T13:33:08.798Z",
"serviceInstanceCount": 0,
"assignedCustomActions": [
    {
        "customActionName": "ind-action",
        "customActionId": "621cca7a218004001ac4bc03",
        "customInputParams": {
            "inputurl": "${svc_instance.output_parameters.todosUrl3}"
        }
    }
],
"id": "621ccec6218004001ac4bc05",
"actions": [
    {
        "type": "provision",
        "name": "Provision",
        "description": "Default action for deployment of service",
        "input_parameters": [],
        "flow": {
            "conditions": [],
            "templates": [],
            "resthooks": [
                {
                    "resthook": {
                        "title": "resthook_cfb688",
                        "id": "resthook00fle7c5",
                        "headers": {
                            "accept": "application/json",
                            "Content-Type": "application/json"
                        },
                        "editable": true,
                        "url": "https://jsonplaceholder.typicode.com/todos/2",
                        "method": "GET"
                    }
                }
            ]
        },
        "notifications": [],

```



```

        "sequence": {
          "0": "resthook00f1e7c5"
        },
        "error_sequence": {}
      },
      "output_parameters": [
        {
          "name": "todosUrl3",
          "label": "todosUrl3",
          "customtype": "string",
          "type": "string",
          "immutable": true,
          "hidden": false,
          "required": false,
          "secured": false,
          "default": "https://jsonplaceholder.typicode.com/todos/3"
        }
      ]
    },
    "plans": [
      {
        "name": "Standard",
        "description": "To deploy a Standard plan",
        "actions": [],
        "plan_parameters": []
      }
    ],
    "tagId": "621c8329218004001ac4bbe2",
    "warning": []
  }
}

```

Additional information for the custom action to a service manifest API

1. You can follow either one of the steps to assign custom input parameters. In this case, `inputurl` values are assigned as shown:
 - a. While assigning custom actions, you can attach the provision action output parameters of a service manifest to the custom action by using interpolation `${svc_instance.output_parameters.<o/p param>}`, which gets resolved when you try to deploy the independent custom action.

Example

```

{
  "custom_action_details": [
    {
      "customActionName": "ind-action",
      "customActionId": "<custom action id>",
      "customInputParams": {
        "inputurl": "${svc_instance.output_parameters.todosUrl3}"
      }
    }
  ]
}

```

- b. You can directly assign the values which overrides the default values at the time of deployment of the custom action.

Example

```

{
  "custom_action_details": [

```

```

    {
      "customActionName": "ind-action",
      "customActionId": "<custom action id>",
      "customInputParams": {
        "inputurl": "http://example.com"
      }
    }
  ]
}

```

2. You can assign custom actions for both publish and draft service manifest.
3. Reassigning custom actions for a service manifest overrides the previous assigned custom actions.

Unlock a custom action

You cannot unlock the custom action until it is unassigned with all the service versions associated with it.

```

curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/CustomAction/<customid>/unlock?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d {}

```

Response

```

{
  "data": {
    "name": "ind-action",
    "ServiceEngine_InputFile": {
      "action": {
        "type": "custom",
        "name": "ind-action",
        "category": "demo-category",
        "description": " Independent custom action",
        "input_parameters": [
          {
            "name": "inputurl",
            "label": "inputpurl",
            "customtype": "string",
            "type": "string",
            "immutable": false,
            "hidden": false,
            "required": false,
            "secured": false,
            "isDynamic": false,
            "default": "https://jsonplaceholder.typicode.com/todos/1"
          }
        ],
        "flow": {
          "conditions": [],
          "templates": [],
          "resthooks": [
            {
              "resthook": {
                "title": "resthook_354d4a",
                "id": "resthook98f44a65",
                "headers": {
                  "accept": "application/json",
                  "Content-Type": "application/json"
                }
              }
            }
          ]
        }
      }
    }
  }
}

```

```

        },
        "editable": true,
        "url": "${input_parameters.inputurl}",
        "method": "GET"
    }
}
],
"notifications": [],
"sequence": {
    "0": "resthook98f44a65"
},
"error_sequence": {}
},
"output_parameters": [
    {
        "name": "resthookOutput",
        "label": "resthookoutput",
        "customtype": "string",
        "type": "string",
        "immutable": true,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "default": "${resthooks.resthook98f44a65.output.body}"
    }
]
}
},
"categoryId": "621cc8e6218004001ac4bc00",
"namespaceId": "cp4aiops",
"tenantId": "cf5c3f7d-a70d-435c-9554-55d8e3441380",
"creationTime": "2022-02-28T13:13:30.918Z",
"lastUpdated": "2022-02-28T13:20:01.513Z",
"status": "Unlocked",
"id": "621cca7a218004001ac4bc03"
}
}

```

Dynamic parameter discovery APIs

This section includes all dynamic parameter discovery related APIs.

- [Clean Cache Data](#)
- [Update clear cache frequency](#)

Clean Cache Data

You must be a cluster administrator or administrator to run this API. Otherwise, an unauthorized error message is displayed.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/dynamicparameterdiscovery/clean-cache?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"

```

Update clear cache frequency

You must be a cluster administrator or administrator to run this API. Otherwise, an unauthorized error message is displayed.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/dynamicparameterdiscovery/update-cache-config?
frequency=10&tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Application APIs

This section includes all application APIs with examples.

- [Creating category for applications](#)
- [Creating an application](#)
- [Importing an application from GitHub repository](#)
- [Pushing an application to GitHub](#)
- [Duplicating an application](#)
- [Editing an application](#)
- [Deploying an application](#)
- [Publishing an application](#)
- [Retrieving application template](#)

Creating category for applications

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppCategories?
ace_orgGuid=${MCM_TEAM}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d {"name": "AppCategorySample"}
```

Result

```
{
  {
    "name": "AppCategorySample",
    "id": "5fc9df7661b3e8001b7c2511",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c"
  }
}
```

Creating an application

```
curl -k -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/createApp?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d @application_create.json
```

application_create.json

```
{
  "data": {
    "name": "ApplicationSample",
    "namespaceId": "abcdef123456",
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\nlabels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n      kind: Subscription\n    - group: core.hybridapp.io\n      kind: Deployable\n  selector:\n    matchLabels:\n      app: nginx-app-details\nstatus: {}",
    "description": "",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "imagepath": "serviceicon_1.svg"
  }
}
```

Response:

```
{
  "data": {
    "name": "ApplicationSample",
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\nlabels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n      kind: Subscription\n    - group: core.hybridapp.io\n      kind: Deployable\n  selector:\n    matchLabels:\n      app: nginx-app-details\nstatus: {}",
    "status": "Draft",
    "description": "",
    "longdescription": "",
    "namespaceId": "abcdef123456",
    "creationTime": "2020-12-04T08:06:00.395Z",
    "lastUpdated": "2020-12-04T08:06:00.395Z",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "inputparameters": [],
    "imagepath": "serviceicon_1.svg",
    "id": "5fc9ede861b3e8001b7c2515"
  }
}
```

Importing an application from GitHub repository

This API call imports an application from the GitHub repository.

```
curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/pullAppModelFromGIT?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @application_import.json
```

application_import.json

```
{
  "app_details": {
    "name": "Application",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "description": "ImportFromGitHub",
    "git_details": {
```

```

        "git_type": "github",
        "git_token": <github_token>,
        "file_path": "mysubdir/application.yaml",
        "git_ref": "master",
        "git_url": "http://mygithub/user/reponame"
    }
}
}

```

Response:

```

{
  "pullAppModelFromGITResponse": {
    "statusCode": 200,
    "status": "success",
    "message": "Application imported successfully"
  }
}

```

Pushing an application to GitHub

This API pushes an application to GitHub.

```

curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/${APP_ID}/pushAppToGit?
ace_orgGuid=${MCM_TEAM}&tenantId=${TSA_TENANT_ID}' \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @push_application.json
push_application.json
{
  "git_details": {
    "git_type": "github",
    "git_token": <github_token>,
    "file_path": "mysubdir/application.yaml",
    "git_ref": "master",
    "git_url": "http://mygithub/user/reponame"
  },
  "commit_details": {
    "commit_message": "Sample file to push an Application"
  }
}

```

Response:

```

{
  "pushAppToGITResponse": {
    "statusCode": 200,
    "status": "success",
    "message": "App pushed successfully"
  }
}

```

Duplicating an application

```

curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/${APP_ID}/duplicateApp?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \

```

```
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d {"app_name": <Duplicate_APP_Name>}
```

Response:

```
{
  "data": {
    "name": <Duplicate_APP_Name>,
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\nlabels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n    kind: Subscription\n    - group: core.hybridapp.io\n    kind: Deployable\n  selector:\n    matchLabels:\n      app: nginx-app-details\nstatus: {}",
    "status": "Draft",
    "description": "",
    "longdescription": "",
    "namespaceId": "abcdef123456",
    "creationTime": "2020-12-04T08:06:00.395Z",
    "lastUpdated": "2020-12-04T08:06:00.395Z",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "inputparameters": [],
    "imagepath": "serviceicon_1.svg",
    "id": "5fc9ede861b3e8001b7c2515"
  }
}
```

Editing an application

This API edits an application. Published applications cannot be edited.

```
curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/${APP_ID}/edit?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @application_edit.json
```

application_edit.json

```
{
  "data": {
    "name": "ApplicationSample",
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\nlabels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n    kind: Subscription\n    - group: core.hybridapp.io\n    kind: Deployable\n  selector:\n    matchLabels:\n      app: ${matchLabel}\nstatus: {}",
    "status": "Draft",
    "description": "",
    "longdescription": "",
    "namespaceId": "",
    "creationTime": "2020-12-04T08:06:00.395Z",
    "lastUpdated": "2020-12-04T08:06:00.395Z",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "inputparameters": [
      {
        "name": "matchLabel",

```

```

        "label": "matchLable",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "isDynamic": false,
        "permission": "Read-Write",
        "default": "nginx-app-details"
    }
],
"imagepath": "serviceicon_1.svg",
"id": <APP_ID>
}
}

```

Response

```

{
  "data": {
    "name": "ApplicationSample",
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\nlabels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n    kind: Subscription\n  - group: core.hybridapp.io\n    kind: Deployable\n  selector:\n    matchLabels:\napp: ${matchLabel}\nstatus: {}",
    "status": "Draft",
    "description": "",
    "longdescription": "",
    "namespaceId": "",
    "creationTime": "2020-12-04T08:06:00.395Z",
    "lastUpdated": "2020-12-04T08:30:23.117Z",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "inputparameters": [
      {
        "name": "matchLabel",
        "label": "matchLable",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "isDynamic": false,
        "permission": "Read-Write",
        "default": "nginx-app-details"
      }
    ],
    "imagepath": "serviceicon_1.svg",
    "id": <APP_ID>
  }
}

```

Deploying an application

This API deploys a published application.


```
curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/${APP_ID}/deploy?
ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d @application_deploy.json
```

application_deploy.json

```
{
  "name": "nginx-app-3",
  "parameters": [
    {
      "name": "matchLabel",
      "value": "nginx-app-details"
    }
  ]
}
```

Response:

```
{
  "instance": {
    "name": "nginx-app-3",
    "status": "Submitted",
    "creationTime": "2020-12-04T08:38:17.933Z",
    "lastUpdated": "2020-12-04T08:38:18.154Z",
    "stack_id": "5fc9f57af86820001b650685",
    "action": "CREATE",
    "logs": "Request queued",
    "namespaceId": "abcdef123456",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "mode": "test",
    "id": "5fc9f57961b3e8001b7c2517",
    "appModelId": <APP_ID>
  }
}
```

Publishing an application

This API publishes an application.

```
curl -k -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel/${APP_ID}/publish?
ace_orgGuid=${MCM_TEAM}&tenantId=${TSA_TENANT_ID}' \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
```

Response:

```
{
  "data": {
    "name": "ApplicationSample",
    "spec": "apiVersion: app.k8s.io/v1beta1\nkind: Application\nmetadata:\n
labels:\n  app: nginx-app-details\n  name: nginx-app-3\n  namespace: ns-sub-
1\nspec:\n  componentKinds:\n    - group: app.ibm.com\n      kind: Subscription\n    -
group: core.hybridapp.io\n      kind: Deployable\n  selector:\n    matchLabels:\n
app: ${matchLabel}\nstatus: {}",
    "status": "Published",
```

```

    "description": "",
    "longdescription": "",
    "namespaceId": "",
    "creationTime": "2020-12-04T08:06:00.395Z",
    "lastUpdated": "2020-12-04T08:43:42.368Z",
    "tenantId": "83d50ee1-849b-4b33-bb25-9cb05f9bd87c",
    "categoryId": "5fc9df7661b3e8001b7c2511",
    "inputparameters": [
      {
        "name": "matchLabel",
        "label": "matchLable",
        "customtype": "string",
        "type": "string",
        "immutable": false,
        "hidden": false,
        "required": false,
        "secured": false,
        "description": "",
        "isDynamic": false,
        "permission": "Read-Write",
        "default": "nginx-app-details"
      }
    ],
    "imagepath": "serviceicon_1.svg",
    "id": <APP_ID>
  }
}

```

Retrieving an application template

This API retrieves the application template details based on given application name.

```

curl -k -X GET \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/AppModel?filter[where]
[name]=${APPNAME}&type=template&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESP
ACE}&tenantId=${TSA_TENANT_ID}' \
-H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'

```

Response:

```

[
  {
    "description": "",
    "name": <APPNAME>,
    "type": "Application",
    "manifest": {
      "template_type": "AppModel",
      "template_format": "JSON",
      "template_provider": "Cloud Pak for Multicloud Management",
      "skip_formatting": true,
      "template": {
        "templateVariables": {
          "template_input_params": [
            {
              "name": "matchLabel",
              "label": "matchLable",
              "customtype": "string",
              "type": "string",
              "immutable": false,
              "hidden": false,

```

```

        "required": false,
        "secured": false,
        "description": "",
        "isDynamic": false,
        "permission": "Read-Write",
        "default": "nginx-app-details"
    },
    ],
    "template_output_params": []
}
},
"metadata": {
    "displayName": "ApplicationSample",
    "longDescription": ""
},
"id": <App_ID>,
"default_template_version": "1.0",
"template_version": {
    "id": <App_ID>,
    "ref": "1.0",
    "name": "1.0",
    "enabled": true,
    "compatible_versions": [
        "1.0"
    ],
    "displayUrl": ""
},
"template_versions": [
    {
        "id": <APP_ID>,
        "ref": "1.0",
        "name": "1.0",
        "enabled": true,
        "compatible_versions": [
            "1.0"
        ]
    }
],
"valid_connections": [],
"provider_engine_versions": null
}
]

```

Shared parameter APIs

This section includes all APIs for shared parameters.

- [Create new Data Type](#)
- [List all Data Types](#)
- [Lookup Data Objects \(implicit or explicit\) for the Data Types specified in the request](#)
- [Resolve the Data Type variables for the template parameters](#)
- [Update Data Type](#)
- [Delete Data Type](#)
- [Create new Data Object](#)
- [Retrieve Data Object](#)
- [Update Data Object](#)
- [Delete Data Object](#)

Create new Data Type

This API creates a new Data Type

```
curl -X POST \
  'https://{TSA_HOSTNAME}/cam/api/v1/datatypes?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
  -H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  --insecure \
  -d '{
    "name": "DemoOneType",
    "label": "Demo One Type",
    "description": "Datatype for DemoOne",
    "attributes": [
      {
        "name": "hostip",
        "default": "0.0.0.0",
        "regex": "[0-9]+\.\.[0-9]+\.\.[0-9]+\.",
        "type": "string",
        "required": "true",
        "secured": "false",
        "label": "Host IP address",
        "hidden": "false",
        "immutable": "false"
      },
      {
        "name": "port",
        "description": "Port number",
        "default": "80",
        "range": {
          "min": "80",
          "max": "9999"
        },
        "type": "counter",
        "required": "true",
        "secured": "false",
        "label": "Port",
        "hidden": "false",
        "immutable": "false"
      },
      {
        "name": "username",
        "default": "admin",
        "description": "User name",
        "hidden": false,
        "immutable": false,
        "label": "User Name",
        "required": true,
        "secured": false,
        "type": "string"
      },
      {
        "name": "password",
        "description": "Password for the host",
        "hidden": false,
        "immutable": false,
        "label": "Password",
        "regex": "[\\w]{8}",
        "required": true,
        "secured": true,
```

```

        "type": "password"
    }
]
}';

```

- **ROLE:**

ADMINISTRATOR

- **Params:**

```

{
  name : [string],
  label: [string],
  description : [string],
  attributes : [
    {
      name: [string],
      default: [string],
      type: [string],
      label: [string],
      required: [boolean],
      hidden: [boolean],
      immutable: [boolean],
      secured: [boolean]
    }
  ]
}

```

- **Success Response:**

- Code: 200 Content: { id : 1, name : "LDAPServer", description: "LDAP server data type", attributes: [...] }

- **Error Response:**

- Code: 400 BAD REQUEST Content: { error : "Data type \${name} already defined." }

Or

- Code: 401 UNAUTHORIZED Content: { error : "You are unauthorized to make this request." }

List all Data Types

This API lists all available Data Types.

```

curl -X GET \
  'https://{TSA_HOSTNAME}/cam/api/v1/datatypes?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
  -H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  --insecure

```

- **ROLE:**

ADMINISTRATOR

- **Success Response:**

- Code: 200 Content: [{ id : 1, name : "LDAPServer", description: "LDAP server data type", attributes: [...] }, ...]

Lookup Data Objects (implicit or explicit) for the Data Types specified in the request

```
curl -X POST \
  'https://{TSA_HOSTNAME}/cam/api/v1/datatypes/lookup?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
  -H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "datatypes" : [
      "DemoOneType"
    ]
  }'
```

- **ROLE:**

tenantMember

- **Params:**

```
{
  datatypes : [datatype_name1, datatype_name2, ...]
}
```

- **Success Response:**

- Code: 200 Content:

Example

```
[
  {
    "name": "datatype_name1",
    "label": "datatype_label1",
    "dataobjects": [
      {
        "id": "1234567890",
        "name": "dataobject name",
        "datatype": "datatype_name1",
        "provider": "IBM"
      }
    ]
  }
]
```

Resolve the Data Type variables for the template parameters

POST /datatypes/resolve

- **ROLE:**

tenantMember

- **Params:**

Example:

```
{
  "input_dataobjects": [
    {
      "id": "1234567890",
      "name": "dataobject name",
      "datatype": "datatype_name1",
      "provider": "IBM"
    }
  ],
  "parameters": {
    "template_input_params": [
      {
        "name": "param1",
        "type": "string",
        "default": "${ContentRuntime.output1}",
      }
    ]
  }
}
```

- **Success Response:**

- Code: 200 Content: { template_input_params: [{ name: 'param1', type: 'string', default: 'output1 value'}]}

Update Data Type

```
curl -X PUT \
  'https://{TSA_HOSTNAME}/cam/api/v1/datatypes/{DATATYPE_ID}?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
  -H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{"description": "updated description"}'
```

- **ROLE:**

ADMINISTRATOR

- **URL params:**

datatype ID or datatype name

- **Params:**

The following parameters are optional, for example, if you only want to update the 'attributes' the request body can contain only the 'attributes'.

```
{
  name : [string],
  label: [string],
  description: [string],
  attributes:[
    {
      name: [string],
      default: [string],
      type: [string],
      label: [string],
      required: [boolean],
      hidden: [boolean],
      immutable: [boolean],
```

```

    secured: [boolean]
  }
]
}

```

- **Success Response:**

- Code: 200 Content: { id : 1, name : "LDAPServer", description: "LDAP server data type", attributes: [...] }

- **Error Response:**

- Code: 400 BAD REQUEST Content: { error : "Data type \${name} already defined." }

Or

- **Code: 401 UNAUTHORIZED Content:** { error : "You are unauthorized to make this request." }

Delete Data Type

```

curl -X DELETE \
  'https://{TSA_HOSTNAME}/cam/api/v1/datatypes/{DATATYPE_ID}?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
-H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'

```

- **ROLE:**

ADMINISTRATOR

- **URL params:**

datatype ID or datatype name

- **Success Response:**

- Code: 201 Content: None

- **Error Response:**

- Code: 404 NOT FOUND Content: { error : "Data type \${id} does not exist." }

Or

- **Code: 401 UNAUTHORIZED Content:** { error : "You are unauthorized to make this request." }

Create new Data Object for specific namespace

This API creates a new Data Object.

```

curl -X POST \
  'https://{TSA_HOSTNAME}/cam/api/v1/dataobjects?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
-H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
--insecure \
-d '{

```



```
"datatype": "DemoOneType",
"name": "DemoOneObject",
"namespaceId": "{{NAMESPACE}}",
"label": "DemoOneObject",
"attributes": [
  {
    "name": "hostip",
    "value": "127.0.0.1"
  },
  {
    "name": "port",
    "value": 587
  },
  {
    "name": "username",
    "value": "user1"
  },
  {
    "name": "password",
    "value": "test1234"
  }
]
}'
```

- **ROLE:**

ADMINISTRATOR

- **Success Response:**

- Code: 200 Content: { id : 1, name : "LDAPServer.production", description: "My LDAP server for production", attributes:[...] }

- **Error Response:**

- Code: 400 BAD REQUEST Content: { error : "Data object \${name} already defined." }

Or

- Code: 401 UNAUTHORIZED Content: { error : "You are unauthorized to make this request." }

Optionally, you can add `cloudOE_spaceGuid=${NAMESPACE}` query parameter to create the Data Object in the namespace that `NAMESPACE` resolves to.

Retrieve Data Object

```
curl -X GET
'https://{TSA_HOSTNAME}/cam/api/v1/dataobjects?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}'
-H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}'
-H 'Content-Type: application/json'
-H 'cache-control: no-cache'
```

- **ROLE:**

ADMINISTRATOR

- **URL params:**

dataobject ID

- **Success Response:**

- Note: The attributes are merged with the datatype's attributes.
- Code: **200** Content:

```
{
  "id" : 1,
  "name": "production",
  "label": "Production LDAP server",
  "datatype": "com.ibm.cam.ldapserver",
  "description": "My LDAP server for production",
  "attributes": [
    {
      "name": "ipaddress",
      "value": "127.0.0.1",
      "default": "",
      "type": "string",
      "required": "true",
      "secured": "false",
      "label": "IP address",
      "hidden": "false",
      "immutable": "false"
    },
    {
      "name": "port",
      "value": "367",
      "default": "",
      "type": "string",
      "required": "true",
      "secured": "false",
      "label": "Port",
      "hidden": "false",
      "immutable": "false"
    }
  ]
}
```

- **Error Response:**

- Code: **400 BAD REQUEST** Content: { error : "Data object \${name} already defined." }
- Or
- Code: **404 NOT FOUND** Content: { error : "Data object \${id} does not exist." }
- Or
- **Code: 401 UNAUTHORIZED** Content: { error : "You are unauthorized to make this request." }

Optionally, you can add `cloudOE_spaceGuid=${NAMESPACE}` query parameter to create the Data Object in the namespace that `NAMESPACE` resolves to.

Update Data Object

```
curl -X PUT \
  'https://{TSA_HOSTNAME}/cam/api/v1/dataobjects/{Dataobjects_ID}?
  ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
  -H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
  -H 'Content-Type: application/json' \
```

```
-H 'cache-control: no-cache' \
-d '{"description": "updated description"}'
```

- **ROLE:**

ADMINISTRATOR

- **URL params:**

dataobject ID

- **Params:**

```
{
  name: [string],
  label: [string],
  datatype: [string],
  description: [string],
  attributes: [
    {
      // attribute definition
      ...
    }
  ]
}
```

- **Success Response:**

- Code: 200 Content: { id : 1, name : "LDAPServer.production", description: "My LDAP server for production", attributes:[...] }

- **Error Response:**

- Code: 400 BAD REQUEST Content: { error : "Data object \${name} already defined." }

Or

- Code: 404 NOT FOUND Content: { error : "Data object \${id} does not exist." }

Or

- **Code: 401 UNAUTHORIZED Content:** { error : "You are unauthorized to make this request." }

Optionally, you can pass `cloudOE_spaceGuid=${NAMESPACE}` in the query parameter to change the namespace of the Data Object. The modified namespace value is the value that **NAMESPACE** resolves to.

Delete Data Object

```
curl -X DELETE \
'https://{TSA_HOSTNAME}/cam/api/v1/dataobjects/{Dataobjects_ID}?
ace_orgGuid=all&cloudOE_spaceGuid=default&tenantId={{TSA_TENANT_ID}}' \
-H 'Authorization: Bearer {{TSA_BEARER_TOKEN}}' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
```

- **ROLE:**

ADMINISTRATOR

- **URL params:**

dataobject ID

- **Success Response:**

- Code: 200 Content: { id : 1, name : "LDAPServer.production", description: "My LDAP server for production", attributes:[...] }

- **Error Response:**

- Code: 404 NOT FOUND Content: { error : "Data object \${id} does not exist." }

Or

- **Code: 401 UNAUTHORIZED Content:** { error : "You are unauthorized to make this request." }

SMTP Connections APIs

Learn more about SMTP connection APIs.

- [Create SMTP connection](#)
- [Modify SMTP connection](#)
- [Test SMTP connection](#)
- [Get list of all SMTP connections](#)
- [Get list of SMTP connections by ID](#)
- [Delete SMTP connection](#)

Create SMTP connection

To use email notifications, you must configure a connection to the SMTP server by running this API. Creating a secondary connection is optional.

1. Create a primary connection.

```
curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create_primary.json \
"https://${TSA_HOSTNAME}/cam/composer/api/v1/smtpconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

POST method

Example request body:

cc_create_primary.json for primary connection:

```
{
  "name": "PRIMARY",
  "connection_parameters": [{
    "name": "hostname",
    "value": "smtp.example.com"
  }, {
    "name": "port",
    "value": 25
  }, {
```

```

        "name": "username",
        "value": "myusername"
    }, {
        "name": "password",
        "value": "mypassword"
    }
  ]
}

```

2. Create a secondary connection.

```

curl -k -X POST \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d @cc_create_secondary.json \
"https://${TSA_HOSTNAME}/cam/composer/api/v1/smtpconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}

```

POST method

Example request body: cc_create_secondary.json for secondary connection

```

{
  "name": "SECONDARY",
  "connection_parameters": [{
    "name": "hostname",
    "value": "test"
  }, {
    "name": "port",
    "value": 25
  }, {
    "name": "username",
    "value": "test"
  }, {
    "name": "password",
    "value": "test"
  }
  ]
}

```

The notification tries to connect to the SMTP server defined in the primary connection. If this connection fails, the notification tries to connect to the SMTP server defined in the secondary connection.

Modify SMTP connection

```

curl -X POST \
'https://${TSA_HOSTNAME}/cam/composer/api/v1/smtpconnections/patchConnectionParametersById?ace_orgGuid=all&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d '{
  "id": "${SMTPCONNECTION_ID}",
  "connection_parameters": [
    {"name": "port", "value": 587},
    {"name": "password", "value": "my-new-password"}
  ]
}'

```

Where **TSA_TENANT_ID** cannot be null.

POST method

The `patchConnectionParameterById` is used to update some of the parameters in the existing SMTP Connection record. Both `patchConnectionParameterById` and `connection_parameters` are mandatory. `TSA_TENANT_ID` cannot be null.

Test SMTP connection

Run this API to test SMTP connection parameters before you create or save a connection.

```
curl -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/smtpconnections/testEmail?
ace_orgGuid=all&tenantId=${TSA_TENANT_ID}' \
  -H 'Authorization: Bearer ${TSA_BEARER_TOKEN}' \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "to": "${SEND-TEST-EMAIL-TO}",
    "conn_params": [
      {"name": "hostname", "value": "${SMTP_HOST}"},
      {"name": "port", "value": "${SMTP_PORT}" },
      {"name": "username", "value": "${USERNAME}"},
      {"name": "password", "value": "${PASSWORD}" }
    ]
  }'
```

Where `TSA_TENANT_ID` cannot be null.

POST method

The `/testEmailById` is similar to `/testEmail` but it uses persisted parameters from a particular record. Optionally, you can send `conn_params` argument whenever you want to use some parameters with changed values.

POST method

```
curl -X POST \
  'https://${TSA_HOSTNAME}/cam/composer/api/v1/smtpconnections/testEmailById?
ace_orgGuid=all&cloudOE_spaceGuid=${NAMESPACE}&tenantId=${TSA_TENANT_ID}' \
  -H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  -H 'Content-Type: application/json' \
  -H 'cache-control: no-cache' \
  -d '{
    "id": "${SMTPCONNECTION_ID}",
    "to": "my-test-email@test.xyz.net",
    "conn_params": [
      {"name": "port", "value": 587},
      {"name": "password", "value": "my-new-password"}
    ]
  }'
```

Where,

- The `conn_params` is optional argument. It can contain subset of parameters that you edit in the user interface. The rest of the parameters are taken from SMTPConnection for the id.
- The `id` and `to` values are mandatory arguments.
- `TSA_TENANT_ID` cannot be null.

Get list of all SMTP connections

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/composer/api/v1/smtptconnections?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}
```

GET method

Get list of SMTP connections by ID

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/composer/api/v1/smtptconnections/<connection-id>?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}
```

GET method

Delete SMTP connection

```
curl -k -X DELETE \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/composer/api/v1/smtptconnections/<connection-id>?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}
```

DELETE Method

Usage specific APIs

As a cluster administrator, you can run the usage specific APIs.

- [Retrieve the current number of deployed virtual machines](#)
- [Retrieve the maximum number of virtual machines for each month](#)
- [Download the managed virtual machines report](#)

You must be a cluster administrator to run these APIs. Otherwise, an unauthorized error message is displayed.

Retrieve the current number of deployed virtual machines

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/api/v1/managedvirtualmachines/countActive?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Retrieve the maximum number of virtual machines for each month

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
  "https://${TSA_HOSTNAME}/cam/api/v1/managedvirtualmachines/highwatermark?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Download the managed virtual machines report

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/managedvirtualmachines/download?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}"
```

Example to retrieve the list of deployed templates

Example API flow to retrieve the list of deployed templates.

A series of API calls to get the list of deployed templates:

```
curl -k -X POST \
-H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8" \
-d
"grant_type=password&username=${TSA_USER}&password=${TSA_USER_PASSWORD}&scope=open
id"\
https://${MCM_CLUSTER_HOSTNAME}/idprovider/v1/auth/identitytoken
```

Example:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded; charset=UTF-8"
-d "grant_type=password&username=admin&password=admin&scope=openid"
https://<MCM_CLUSTER_HOSTNAME>/idprovider/v1/auth/identitytoken
```

Response:

```
{
  "access_token": "gvWTQGcb9DzZZb5Oj6GYRjM00OUBYTCxgjJULI0dDURsw6p2uWdG3jDCe8S8UCqLVS
oU3wwr9toWKGtISkCmrTtBBGzET8hDhBLd3DXU1PMt1jNZAX65aRLvEhLHouVc1H1WQVavVYT9RBehfNct
kCbYsaToh8rWmzwxAQsN5vRESk727VeTapSfHstx0CL2oUNanHuu73hukWQ
I3MEsN2X9q1gnGIyUdeSdHQZYJ1G1l18h0GV9vNzA48dgoEPSptbGK3vRt3p2gK4lzzzwjjPiRX8LAqPXYx
jc3GN4RENkhF2XhHLSZJqWAni9ViRhjQiRTOZjpTLu8xdYg4HPwxqywBviIhUVPYqDe2CaHqjl2JVzeVoo
dByY7rsSEXiZviwjrVd6vcoQ7nLLkQ4JrpxZP7Zm1SKluN00I7ZolGz1gk81
Y1F8OKISZSEyp7PZnHKI6RwNiFtWc54xvAh8t1Q15NH1F3ZsyTOa21IKFJeO1SW8ejMLQaUxfS5uVIgHn
OetY4auLU7X82IaR9r6n2BB9ftUWq7ZKf2uG1pC1MwW5p1oyNvulsCrgh7bvsthwGR1OdRp04ze94rmiDi
hnCGCULcf6hWxKNq8cqVsLmk5IJKfplVxu2OYFWEMd0bqL1YzaDpdgtDD5iH
0BNw83d2omjisukkfINNA0T0Z7vpuxuXZ13NBC7fsK1FvN1WWDDs7oHcZ8SPYsITu4cbdjuVayilE1QCAx
TzyDVz70DQa84qvl1sqczn5jPGVh5cyBvikTEPD6BEQuBhV7Tbq2WVhuS3IuAHWUqfQb085Xb14SqQOXRta
tlcRZb11npg44WLXTqH4Hl1glpyQpbAJTWGCV5AgDWc8fGjjUIz6nWo50700mr
cvNOhVKzshXv49PqaONdLQpQ13LnURCsD9L3IpdJfxojR6eCSG4qQW1JVJC8uSQtjKYkqvdlmAPrxUgMv
8yrPfIEOH1tV6h45SricSYKoOfBG72DqIXH", "token_type": "Bearer", "expires_in": 43199, "sco
pe": "openid", "refresh_token": "fvNZN6AjYbS3RDXYi0rchBpbcezeMCJT5
BmPEEHomC529rr0Bx", "id_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdF9oYXNoIjI
oiMnhka3hoaDY5OTdpdHE2bmFsZHMlLCJyZWZsbU5hbWUuOiJjdXN0b21SZWFsbSIzInVuaXF1ZVN1Y3Vy
aXR5TmFtZSI6ImFkbWluIiwiaXNzIjoiaHR0cHM6Ly9teWNsdXN0ZXIuaWNwOjE
0NDMvbm2lkYy9lbmRwb2ludC9PUCIsImF1dG8iOiJmZmMzNGJlMjVjNTQ1MDlmMDA3MjRiYjAxOTI2MGZhNzJlI
iwiZXhwIjoxNTA5NzIxMjU5LCJpYXQiOiJlMDk3MzEyNTksInN1YiI6ImFkbWluIn0. tp7MKObvsAqzdAR
Wf3U24Us_f3ZtDjGWl2IC2vYUisUTwKrhC7thoPSFViUmbpOhbswtYXNuGxR2E6E
KsnaU7TDQhrbZdLWWB2E8OC-Kzd6ITEeCmsV8RrWFQmPm5uidCTBmkbSoImcQ2I6dqtD66xiWc0Gq-
sWceVrDL63VpuUbQxuJfmmCswEcDVybpJBOM2xHvGkhkLxwU9pRtnvtTbC1BoMNsIzZn4gR3vpm1ViIMS6
_FVvFuzAHnZ_95fYZ1RTVXr5QImuN4vC-JIZb8KCxLqHvM9GUHus_XaQnQ3npBRXpLluL
YI29yX4VqQNTbS_PKWaolHKGcFNSqLRS8w"
}
```

Note: Save the `access_token` from the response and use it as a bearer token in following API calls.


```
export
TSA_BEARER_TOKEN=gvWTQGcb9DzZZb5Oj6GYRjM0OUBYTcxgjJULI0dDURsw6p2uWdG3jDCe8S8UCqLV
SoU3wwr9toWKGtISkcMrttBBGzET8hDhBLd3DXULPMt1jNZAX65aRLvEhLHouVc1H1WQVavVYT9RBeHfNc
TkCbYsaToh8rWmzwxAQsN5vRESk727VeTapSfHstx0CL2oUNanHuu73hu
kWQI3MEsN2X9q1gnGIyUdeSdHQZYJ1G1l18h0GV9vNzA48dgoEPSPtbGK3vRt3p2gK4lzzzwjjPiRX8LAqP
XYxjc3GN4RENkhF2XhHLSZJqWAni9ViRhjQiRTOZjpTLu8xdYg4HPwxqywBviIhUVPyqDe2CaHqjl2JVze
VoodByY7rsSEXiZviiwjrVd6vcoQ7nLLkQ4JrpdXZP7Zm1SKluNQ0I7ZolGz1gk81
Y1F8OKISZZSEyp7PznHki6RwniFtWc54xvAh8t1Q15NH1F3ZsyTOa21IKFJeO1SW8ejMLQaUxfS5uVIgHn
OetY4auLU7X82Iar9r6n2BB9ftUWq7ZKf2uG1pC1MwW5ployNvulsCrgh7bvsthWGR1OdRp04ze94rmiDi
hnCGCULcf6hWxKNq8cqVsLmk5IJKfplLVxu2OYFWEMd0bqL1YzaDpdgtDD5iH0BN
w83d2omjisukkfINNA0T0Z7vpuxuXZ13NBC7fsK1FvN1WWDDs7oHcZ8SPYsITu4cbdjuVayilE1QCAxTzy
DVz70DQa84qvl5sqczn5jPGVh5cyBvikTEPd6BEQuBhV7Tbq2WVhuS3IuAHWUqfQb085Xb14SsqQOXrtatlC
RZb11npg44WLXTqH4HliglpyQpbAJTWGCV5AgDWc8fGjjUIz6nWo50700mrcvNOh
VKzshXv49PqaONdLQpQ13LnURCsD9L3IpdJfxojR6eCSG4qQW1JVJC8uSQtjKYkqvdlmAPrxUgMv8yrPf
IEOH1tV6h45SricSYKoOfBG72DqIXH
```

```
curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
https://${TSA_HOSTNAME}/cam/tenant/api/v1/tenants/getTenantOnPrem
```

Example:

```
curl -k -X GET -H "Authorization: Bearer
gvWTQGcb9DzZZb5Oj6GYRjM0OUBYTcxgjJULI0dDURsw6p2uWdG3jDCe8S8UCqLVSoU3wwr9toWKGtISk
cMrttBBGzET8hDhBLd3DXULPMt1jNZAX65aRLvEhLHouVc1H1WQVavVYT9RBeHfNcTkCbYsaToh8rWmzwx
AQsN5vRESk727VeTapSfHs
tx0CL2oUNanHuu73hukWQI3MEsN2X9q1gnGIyUdeSdHQZYJ1G1l18h0GV9vNzA48dgoEPSPtbGK3vRt3p2g
K4lzzzwjjPiRX8LAqPXyXjc3GN4RENkhF2XhHLSZJqWAni9ViRhjQiRTOZjpTLu8xdYg4HPwxqywBviIhU
VPyqDe2CaHqjl2JVzeVoodByY7rsSEXiZviiwjrVd6vcoQ7nLLkQ4JrpdXZP7Zm1
SKluNQ0I7ZolGz1gk81Y1F8OKISZZSEyp7PznHki6RwniFtWc54xvAh8t1Q15NH1F3ZsyTOa21IKFJeO1S
W8ejMLQaUxfS5uVIgHnOetY4auLU7X82Iar9r6n2BB9ftUWq7ZKf2uG1pC1MwW5ployNvulsCrgh7bvsth
wGR1OdRp04ze94rmiDihnCGCULcf6hWxKNq8cqVsLmk5IJKfplLVxu2OYFWEMd0b
qL1YzaDpdgtDD5iH0BNw83d2omjisukkfINNA0T0Z7vpuxuXZ13NBC7fsK1FvN1WWDDs7oHcZ8SPYsITu4
cbdjuVayilE1QCAxTzyDVz70DQa84qvl5sqczn5jPGVh5cyBvikTEPd6BEQuBhV7Tbq2WVhuS3IuAHWUqfQ
b085Xb14SsqQOXrtatlCRZb11npg44WLXTqH4HliglpyQpbAJTWGCV5AgDWc8fGjj
UIz6nWo50700mrcvNOhVKzshXv49PqaONdLQpQ13LnURCsD9L3IpdJfxojR6eCSG4qQW1JVJC8uSQtjKY
kqvdlmAPrxUgMv8yrPfIEOH1tV6h45SricSYKoOfBG72DqIXH"
https://9.37.194.250/cam/tenant/api/v1/tenants/getTenantOnPrem
```

Response:

```
{"id":"5c0e0a5e-d431-468c-9a11-a3df3978e424","name":"ICP
Account","ownerId":"On_Prem","wlpClientId":"a71LLVER8bTekvyNmJjEdWY6XO7NGb3I","nam
espaces":[{"name":"default","uid":"default"}, {"name":"kube-public","uid":"kube-
public"}, {"name":"kube-system","uid":"kube-system"},
{"name":"services","uid":"services"}]}
tenantId = response.id = 5c0e0a5e-d431-468c-9a11-a3df3978e424
export TSA_TENANT_ID=5c0e0a5e-d431-468c-9a11-a3df3978e424
MCM team (ace_orgGuid) = all
export MCM_TEAM=all
namespace (cloudOE_spaceGuid) = response.namespaces[0].uid = default
export NAMESPACE=default

curl -k -X GET \
-H "Authorization: Bearer ${TSA_BEARER_TOKEN}" \
"https://${TSA_HOSTNAME}/cam/api/v1/stacks?
tenantId=${TSA_TENANT_ID}&ace_orgGuid=${MCM_TEAM}&cloudOE_spaceGuid=${NAMESPACE}"
```

Example:

```
curl -k -X GET -H "Authorization: Bearer
gvWTQGcb9DzZZb5Oj6GYRjM0OUBYTcxgjJULI0dDURsw6p2uWdG3jDCe8S8UCqLVSoU3wwr9toWKGtISk
cMrttBBGzET8hDhBLd3DXULPMt1jNZAX65aRLvEhLHouVc1H1WQVavVYT9RBeHfNcTkCbYsaToh8rWmzwx
AQsN5vRESk727VeTapSfHstx0CL2oUNanH
```

```
uu73hukWQI3MEsN2X9q1gnGIyUdeSdHQZYJ1G1l8h0GV9vNzA48dgoEPSptbGK3vRt3p2gK4lzzzwjjPiR
X8LAqPXYxjc3GN4RENkhF2XhHLsZJqWAni9ViRhjQiRTOZjpTLu8xdYg4HPwxqywBviIhUVPyqDe2CaHqj
l2JVzeVoodByY7rsSEXiZviiwjrVd6vcoQ7nLlKQ4JrpdXZP7Zm1SKluNQ0I7ZolGz1gk81Y1F8OK
ISZZSEyp7PZnHKI6RwNiFtWc54xvAh8t1Q15NH1F3ZsyTOa21IKFJeO1SW8ejMLQaUxfS5uVIgHnOetY4a
uLU7X82IaR9r6n2BB9ftUWq7ZKf2uG1pC1MwW5ployNvu1sCrgh7bvsthWGRlOdRp04zE94rmiDihNCGCU
lcf6hWxKNq8cqVsLmk5IJKfPbLVxu2OYFWEmd0bqL1YzaDpdgtDD5iH0BNw83d2omjisukkfINNA
0T0Z7vpuxuXZ13NBC7fsK1FvN1WDDs7oHcZ8SPYsITu4cbdjvVayile1QCAxTzyDVz70DQa84qv1sqczn
5jPGVh5cyBVikTEPd6BEQuBhV7Tbq2WVhuS3IuAHWUqfQb085Xb14SqQOXrtatlCRZb11npg44WLXTqH4H
liglpyQpbAJTWGCV5AgDWc8fGjjUIz6nWo5070OmrcvNOhVKzshXv49PqaONdLQpQ13LnUrRCsD9L
3IpdJfxojR6eCSG4qQW1JVJC8uSQtjKYkqvdlmAPrxUgMv8yrPfIEOH1tV6h45SricSYKoOfBG72DqIXH"
https://9.30.37.11/cam/api/v1/stacks?tenantId=5c0e0a5e-d431-468c-9a11-
a3df3978e424&ace_orgGuid=all&cloudOE_spaceGuid=default
```

Response:

```
[{"name": "MyvSphereDeploy", "created_at": "2017-11-02T18:08:00.818Z", "templateId": "59f8b7b50377070020bdf501", "namespaceId": "default",
"templateName": "SingleVirtualMachine-
Hugh", "templateCategory": "userCreated", "templateProvider": "VMware
vSphere", "template_type": "Terraform", "template_format": "HCL", "applied_state_id": "5
9fb5f01873a17001f1d3896", "latest_state_id": "59fb5f01873a17001f1d3896", "cloud_conne
ction_ids":
["59f8b79e0377070020bdf4fe"], "id": "59fb5f00873a17001f1d3895", "tenantId": "5c0e0a5e-
d431-468c-9a11-
a3df3978e424", "status": "IN_PROGRESS", "message": "", "action": "CREATE", "stack_id": "c3
6cc6d0-bff8-11e7-ac54-
5bb358fd234d", "applied_status": "IN_PROGRESS", "applied_action": "CREATE"}]
```

Sample Python script

This section has a sample python script for reference.

```
import requests
import json

print("Managed services APIs - Demo Python Script")

protocol='https://'
tsaHostname='tsa_hostname'
mcmClusterHostname='MCM_CLUSTER_HOSTNAME'
user='TSA_admin_id'
pw='TSA_admin_password'

tsaEndpoint=protocol + tsaHostname
mcmEndpoint=protocol + mcmClusterHostname

<!--GET access_token from IBM Cloud Pak for AIOps-->

postdata = "grant_type=password&username=" + user + "&password=" + pw +
"&scope=openid"
head = {'Content-Type': 'application/x-www-form-urlencoded',
'Accept': 'application/json', "charset" : "UTF-8"}
ret = requests.post(mcmEndpoint + "/idprovider/v1/auth/identitytoken",
                    data=postdata,
                    headers=head,
                    verify=False)
access_token=ret.json()["access_token"]
print("access_token", access_token)

<!-- GET tenantId, team and namespace from Managed services-->
```

```

head = {"Authorization": "bearer " + access_token}
ret = requests.get(tsaEndpoint + "/cam/tenant/api/v1/tenants/getTenantOnPrem",
                  headers=head,
                  verify=False)
tenantId=ret.json()["id"]
namespace=ret.json()["namespaces"][0]["uid"]
if user != 'admin':
    team=ret.json()["namespaces"][0]["teamId"]
else:
    team='all'
print("tenantId", tenantId)
print("team", team)
print("namespace", namespace)

<!-- GET cloudconnections from Managed services-->

parameters={"tenantId":tenantId, "ace_orgGuid":team}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/api/v1/cloudconnections",
                  headers=head,
                  params=parameters,
                  verify=False)
cloudConnections=json.dumps(ret.json())
print("cloudConnections", cloudConnections)

<!-- GET stacks from Managed services-->

parameters={"tenantId":tenantId, "ace_orgGuid":team,
"cloudOE_spaceGuid":namespace}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/api/v1/stacks",
                  headers=head,
                  params=parameters,
                  verify=False)
stacks=json.dumps(ret.json())
print("stacks", stacks)
<!-- GET templates from Managed services -->
parameters={"tenantId":tenantId, "ace_orgGuid":team}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/api/v1/templates",
                  headers=head,
                  params=parameters,
                  verify=False)
templates=json.dumps(ret.json())
print("templates", templates)
<!-- GET templates from Managed services with name filter-->
parameters={"tenantId":tenantId, "ace_orgGuid":team, "filter":'{"where": {"name":
"SingleVirtualMachine" }}'}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/api/v1/templates",
                  headers=head,
                  params=parameters,
                  verify=False)
templates=json.dumps(ret.json())
templateId=ret.json()[0]["id"]
print("templates", templates)
print("templateId", templateId)
<!-- GET sharedruntimeinstances from Managed services-->
parameters={"tenantId":tenantId, "ace_orgGuid":team,
"cloudOE_spaceGuid":namespace}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/api/v1/sharedruntimeinstances",
                  headers=head,

```

```

        params=parameters,
        verify=False)
sharedruntimeinstances=json.dumps(ret.json())
print("sharedruntimeinstances", sharedruntimeinstances)
<!-- GET order from Managed services-->
parameters={"tenantId":tenantId, "ace_orgGuid":team,
"cloudOE_spaceGuid":namespace}
head = {"Authorization": "bearer " + access_token, 'Accept':'application/json'}
ret = requests.get(tsaEndpoint + "/cam/composer/api/v1/ServiceInstances",
        headers=head,
        params=parameters,
        verify=False)
order=json.dumps(ret.json())
print("order", order)

```

Managed services Content APIs

The Managed services Content APIs manage cookbooks and roles in the pattern manager.

The Managed services Content APIs are used to list, update, and delete content on the pattern manager. The pattern manager interacts with the Chef server and software repository to manage cookbooks, recipes, and roles in your Content Runtime environment. When deploying a content runtime, you can install a new Chef server or bring your own; the pattern manager will interact with either. Managed services has a set of middleware cookbooks that can be loaded automatically when deploying the Content Runtime environment or can be loaded manually after the Content Runtime has been created. Load cookbooks, recipes, and roles from IBM, from your own repositories, or from Chef Supermarket.

Access to these APIs is granted with either the access token or the admin token. The access and admin tokens were specified when deploying the Content Runtime. Any API that accepts the access token also accepts the admin token. When using these APIs, the access or admin token must be specified with an authorization HTTP header. For example,

```

# curl -H "Authorization: Bearer $ACCESS_TOKEN"
https://pattern_manager:5443/v1/info/chef

```

Image management API data

API Version

1.0.0

API URI components

Scheme

HTTPS

Port number

5443

Command output format

application/json

Upload cookbooks and roles

Uploads cookbooks from a git repository, chef supermarket, or local repository to the pattern manager.

`POST /v1/upload/chef`

Description

Uploads cookbooks from a git repository, Chef supermarket, or local repository to the pattern manager. Valid sources are the chef supermarket, github.com, gitlab.com, and local repository on the pattern manager file system. To ensure the latest updates are available on your pattern manager, call this API every time there is an update to the cookbooks and roles in your repository.

The pattern manager access token is required to perform this request.

Directory Structure

For cookbooks that are loaded from a git or local repository, it is assumed that they are in a folder named cookbooks. This cookbooks folder can either be at the root directory of your repository or in a folder named Chef that is at the root directory of your repository. Similarly, it is assumed that roles are in a folder named roles, which is either at the root directory of your repository or in a folder named chef within the root directory of your repository.

Directory Structure Example

The following code block depicts an example directory structure:

```
.
|-- github.ibm.com
    |-- example_org
        |-- example_repository
            |-- chef
                |-- cookbooks
                    |-- README.md
                    |-- attributes.json
                    |-- components.json
                    |-- cookbook_metadata.json
                    |-- metadata.rb
                    |-- recipes.json
                |-- roles
                    |-- create_user.json
                    |-- database_v10_install.json
                    |-- database_v15_install.json
```

Parameters

The following table lists the parameters that are used to upload cookbooks and roles to the pattern manager:

Type	Name	Description	Schema	Default	Required
Body	authorization	JSON object containing a personal_access_token property	JSON object		Yes for git repository

Type	Name	Description	Schema	Default	Required
Body	cookbook_name	Name of the cookbook to load when uploading from the chef supermarket. Only valid when setting source_repos to chef_supermarket	String		Yes, when using chef supermarket
Body	cookbook_version	Version of the cookbook to load when uploading from the chef supermarket. Only valid when setting source_repos to chef_supermarket	String		No
Body	deps	JSON array of dependencies required for the cookbook being installed from the chef supermarket. Each item must contain the cookbook_name and an optional cookbook_version parameter. Only valid when setting source_repos to chef_supermarket	JSON array		No
Body	personal_access_token	GitHub personal access token	String		Yes for git repository
Body	cookbooks	States whether cookbooks should be uploaded	Boolean	True	No

Type	Name	Description	Schema	Default	Required
Body	org	Organization where your repositories reside. For GitLab, this is your group name	String		Yes for git repository
Body	branch	GitHub branch for your repositories	String	master	No
Body	repos	For git or local repositories, the regular expression to filter repositories (GitHub) or projects (GitLab) where you want to search for cookbooks and roles	String	.*	No
Body	repos_hostname	The hostname or IP address of your git repository if it does not match the value in source_repos	String	Value of source_repos	No
Body	repos_protocol	The protocol (http or https) used by your git repository	String	https	No
Body	roles	States whether roles should be uploaded	Boolean	False when source_repos is set to chef_supermarket, True otherwise	No
Body	source_repos	Hostname of the git repository. Valid options include chef_supermarket, github.com, gitlab.com, or a path to your local repository in file URI scheme.	String		Yes

Example

The following example shows the request that would be used to import all cookbooks and roles from the preceding example directory structure.

```
POST /v1/upload/chef
```

```
Header: Authorization = Bearer $ACCESS_TOKEN
```

```
{
  "authorization": {
    "personal_access_token": "38eab5732f7411bedb976afcc90a25432d798bda"
  },
  "cookbooks": "True",
  "source_repos": "github.com",
  "org": "example_org",
  "repos": "example_.*",
  "roles": "True"
}
```

This example shows a request to import from a GitLab repository that is hosted somewhere other than the gitlab.com. The value specified for org is the GitLab group name and the repos regular expression will filter projects belonging to that group.

```
POST /v1/upload/chef
```

```
Header: Authorization = Bearer $ACCESS_TOKEN
```

```
{
  "authorization": {
    "personal_access_token": "7411bedb9738eab5732f6afc32d798bdac90a254"
  },
  "cookbooks": "True",
  "source_repos": "gitlab.com",
  "org": "production_group",
  "repos": "cookbooks_.*",
  "roles": "True",
  "repos_hostname": "gitlab-pc.ibm.com",
  "protocol": "http"
}
```

This example shows a request to import the Java cookbook from the chef supermarket. The Java cookbook is dependent on the four other cookbooks listed in the **deps** parameter. If a cookbook in the **dep** array already exists on the Chef server and the **overwrite_existing** parameter is set to true, then it gets overwritten.

```
POST /v1/upload/chef
```

```
Header: Authorization = Bearer $ACCESS_TOKEN
```

```
{
  "cookbook_name": "java",
  "cookbook_version": "1.50.0",
  "source_repos": "chef_supermarket",
  "deps": [
    {
      "cookbook_name": "apt",
      "cookbook_version": "6.1.4"
    },
    {
      "cookbook_name": "homebrew",
      "overwrite_existing": "true"
    }
  ]
}
```



```
    },  
    {  
      "cookbook_name": "windows"  
    },  
    {  
      "cookbook_name": "ohai"  
    }  
  ]  
}
```

This example shows a request to import cookbooks and roles from the pattern manager local filesystem.

POST /v1/upload/chef

Header: Authorization = Bearer \$ACCESS_TOKEN

```
{  
  "cookbooks": "True",  
  "roles": "True",  
  "source_repos": "file:///var/local_repo"  
}
```

Responses

The following table lists responses that are returned for a request to upload cookbooks and roles to the pattern manager:

HTTP Code	Description	Schema
200	OK	No content
400	Bad Request	
403	The requester has not specified the correct access token in the Authorization HTTP header	
500	Internal server error	

List cookbooks and roles

List the cookbooks and roles that have been uploaded to the pattern manager

GET /v1/info/chef

Description

Return a list of all cookbooks and roles that have been uploaded to the pattern manager

Responses

The following table lists API responses:

HTTP Code	Description	Schema
200	Returns the list of cookbooks and roles that reside on the pattern manager.	See example

HTTP Code	Description	Schema
403	The requester has not specified the correct access token in the Authorization HTTP header.	
500	An internal error occurred while processing the request.	

Example

The following block is an example of a successful request:

Header: Authorization = Bearer \$ACCESS_TOKEN

```
{
  "cookbooks": [
    "db2 0.1.36",
    "httpd 1.1.27",
    "ibm_cloud_utils 0.1.29"
  ],
  "recipes": [
    "db2",
    "db2::cleanup",
    "db2::create_db",
    "db2::fixpack",
    "db2::gather_evidence",
    "db2::harden",
    "db2::install",
    "db2::license",
    "db2::prereq",
    "db2::prereq_check",
    "httpd",
    "httpd::cleanup",
    "httpd::config_httpd_conf",
    "httpd::config_ssl",
    "httpd::config_vhosts",
    "httpd::gather_evidence",
    "httpd::install",
    "httpd::prereq",
    "httpd::service",
    "ibm_cloud_utils",
    "ibm_cloud_utils::gather_evidence",
    "ibm_cloud_utils::install",
    "ibm_cloud_utils::prereq"
  ],
  "roles": [
    "db2_create_db",
    "db2_v105_install",
    "db2_v111_install",
    "httpd24-base-install",
    "httpd24-ssl-vhosts"
  ]
}
```

Clear cookbooks and roles

Clears cookbooks and roles from the pattern manager.

POST /v1/clear_env/chef

Description

Deletes cookbooks and roles from the pattern manager. You can specify to delete cookbooks, roles, or both. By default all cookbooks and roles are deleted.

This API requires the pattern manager admin token.

Parameters

The following table lists the parameters that are used to clear cookbooks and roles from the pattern manager:

Type	Name	Description	Schema	Default	Required
Body	cookbooks	States whether cookbooks should be deleted	Boolean	True	No
Body	pattern	Delete cookbooks and roles that match this pattern	String	•	No
Body	roles	States whether roles should be deleted	Boolean	True	No

Example Request

```
POST /v1/clear_env/chef
```

```
Header: Authorization = Bearer $ADMIN_TOKEN
```

```
{
  "cookbooks": "true",
  "roles": "true",
  "pattern": "db2.*"
}
```

Responses

The table lists responses that are returned for a request to clear cookbooks and roles from the pattern manager:

HTTP Code	Description	Schema
200	OK	No content
400	Bad Request	
403	The requester has not specified the correct access token in the Authorization HTTP header	
500	Internal server error	

Get service logs

Return the pattern manager service logs

GET /v1/admin/log

Description

Return the pattern manager service logs in one of three formats. The format returned is based on the value of the Accept HTTP header, with application/json being the default response format.

This request requires the pattern manager admin token. The following table lists the accept HTTP header options:

Accept Header Value	Description
application/json	Filtered service log entries in JSON format
application/gzip	All service logs in a tar.gz file
application/zip	All service logs in a zip file
text/plain	The latest service log file

Parameters

These parameters are only valid when requesting a response in JSON format. If no parameters are specified, then the last 5000 entries in the service log file will be returned.

The following table lists the parameters that are used to filter service log entries returned in JSON response:

Table 2.

Type	Name	Description	Schema	Default	Required
Body	contains	Returns all log entries that contain the specified value	String		No
Body	end_dts	Date timestamp to end log entries collections. Accepted formats include YYYY-mm-dd HH:MM:SS,fff, YYYY-mm-dd HH:MM:SS, and YYYY-mm-dd	String		No
Body	levels	List of logging levels to include: CRITICAL, ERROR, WARNING, INFO, and DEBUG	Array		No

Type	Name	Description	Schema	Default	Required
Body	request_id	Returns entries associated with a specific request ID. Can be used to narrow logs down to a single request train	String		No
Body	remote	List of properties to remove from the JSON response including body, dts, level and request_id	Array		No
Body	start_dts	Date timestamp to start log entries collections. Accepted formats include YYYY-mm-dd HH:MM:SS,fff, YYYY-mm-dd HH:MM:SS, and YYYY-mm-dd	String		No

Example Request

Header: Authorization = Bearer \$ADMIN_TOKEN

```
{
  "contains": "/v1/info/chef",
  "levels": ["DEBUG"],
  "start_dts": "2017-08-03 14:50:10"
}
```

Responses

The following table lists the API responses:

HTTP Code	Description	Schema
200	Returns the requested pattern manager service log entries	
403	The requester has not specified the correct admin token in the Authorization HTTP header.	
500	An internal error occurred while processing the request.	

Example Response

The following block is an example of a successful response in JSON format.

```

{
  log: [
    {
      body: "API response: Status = 200. Data log suppressed***** '/v1/info/chef
GET ",
      dts: "2017-08-03 14:50:10,800",
      level: "DEBUG",
      request_id: "73fdad0f5f4d43829da05ad245e168da"
    },
    {
      body: "***** '/v1/info/chef GETExecuting cmd --> cd /tmp/tmpgkDyxz; knife
ssl fetch ",
      dts: "2017-08-03 14:50:19,286",
      level: "DEBUG",
      request_id: "f7609ed9232548698c1f7c3c82dac9e7"
    }
  ]
}

```