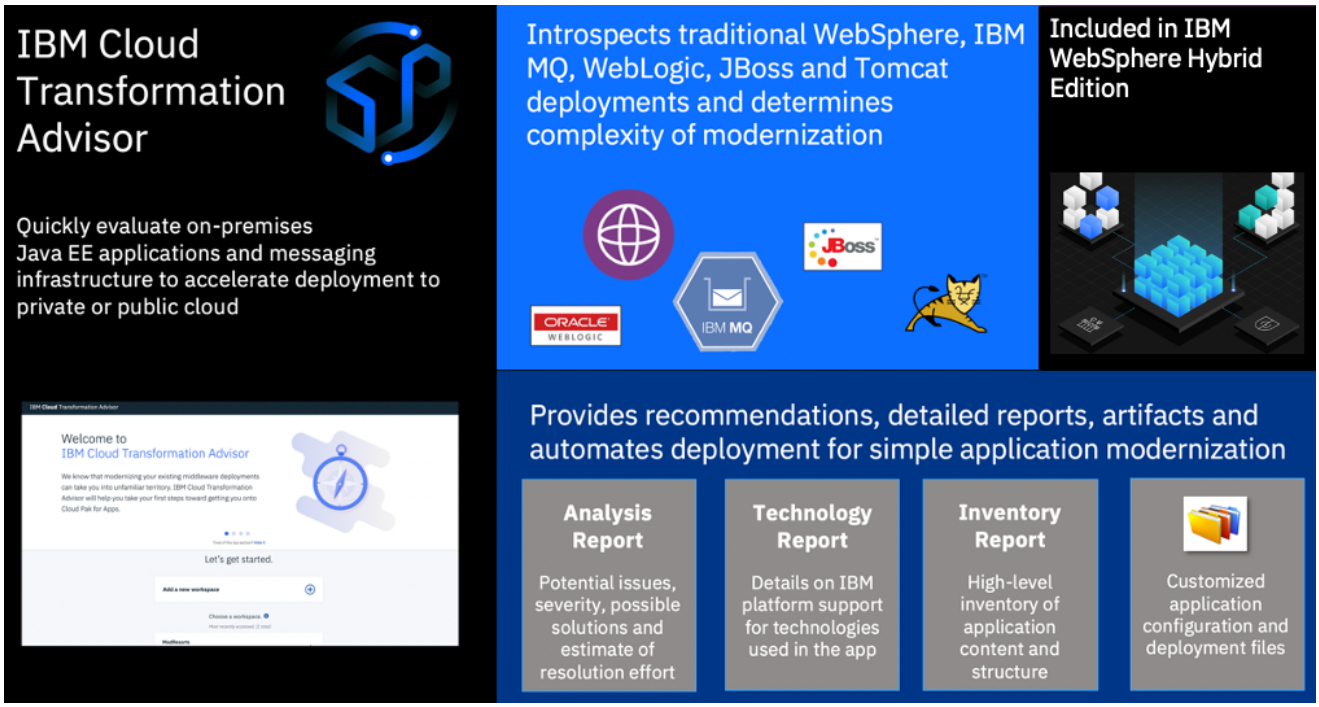
IBM Cloud Transformation Advisor

Tables of Contents

IBM Cloud Transformation Advisor

Welcome to the IBM Cloud Transformation Advisor documentation, where you can find information about how to install, use, and troubleshoot Transformation Advisor.

Transformation Advisor is a great tool that helps businesses modernize and migrate their applications from on-premises environments to the cloud. This process typically requires a lot of preparation and an in-depth analysis of the applications being migrated. Every application is different in terms of how up-to-date it is and how suitable it is for a cloud environment. Users first need to spend time determining the structures of their application, which parts need to be modernized, and which tools and conditions are needed in order to migrate it to the cloud. Transformation Advisor helps with discovering, assessing and migrating your applications.



What's New?

Version 3.10.1 (August 2024)

- Security
  - IBM Cloud Transformation Advisor has addressed multiple security vulnerabilities including those in Node.js, Java SE and various other libraries

Version 3.10.0 (June 2024)

- Migration bundle enhancement
  - Migration bundle will now contain the full set of modernization reports, updated metadata and updated recipe configurations

---

Version 3.9.0 (March 2024)

- Migration bundle enhancement
  - Migration bundle will now contain recipe configuration for automated code remediation
- Easy access to rule help
  - You don't need to open analysis report to see the rule help, it is now shown in the application details page
- Playbook improvements
  - We expanded modernization playbook with new real life scenarios for secure connections and clustered deployments. You can access the playbook here
- Data Collector
  - The Data Collector now uses binary scanner v24.0.0.1

Version 3.8.2 (February 2024)

- Security
  - IBM Cloud Transformation Advisor has addressed multiple security vulnerabilities including those in Node.js, Java SE and various other libraries

---

Version 3.8.1 (January 2024)

- Upload defect
  - Fixed the issue introduced in 3.8.0 where collections with certain dependencies were not uploading
- Automated Code Changes
  - Automated code fix configuration for an application is now filtered to only the required set of fixes
- Security
  - IBM Cloud Transformation Advisor has addressed multiple security vulnerabilities including those in Node.js, Java SE and various other libraries

---

Version 3.8.0 (December 2023)

- Automated Code Changes
  - Automated fixes are provided to automatically update your source code for some issues. For each application we indicate the degree of automation available and the configuration required to execute the updates. Full details on our automated fixes can be found in this documentation
- Data Collector
  - The ability to collect data for only specified applications has been extended to include Tomcat, WebLogic and JBoss
  - The Data Collector now uses binary scanner v23.0.0.5

Version 3.7.1 (October 2023)

- User Experience
  - Provide a more useful and accurate description of development costs
- Security
  - Add back logout button
  - Address multiple security vulnerabilities including those in Node.js, Java SE and various other libraries

---

3.7.0 (September 2023)

- Modernization Guidance
  - Modernization Guidance is a recommended sequence for modernizing your applications so that you build momentum as quickly as possible. The easiest applications that also help as many other applications as possible are prioritized. Using this guidance you will actually see that applications can move from Moderate to Simple based solely on work you have already completed!
- Playbook
  - We now provide an online modernization playbook that gives a full decision tree for your modernization journey. It will take you from your initial collection and assessment all the way through to deployment, showing best practice, tips, tricks and options along the way. You can find the link it in the new Transformation Advisor header, or just click here
- Data Collector
  - The Data Collector now uses binaryScanner v23.0.0.3

3.6.2 (August 2023)

- User Experience
  - The banner indicating that a new version is available will now vanish once that new version is installed
- Security Fixes
  - IBM Cloud Transformation Advisor has addressed multiple security vulnerabilities including those in Node.js, Java SE and various other libraries

---

3.6.1 (July 2023)

- Security Fixes
  - Fix various security issues

3.6.0 (June 2023)

- Migration Plan
  - The migration bundle now extracts all Liberty configuration parameters and sensitive data variables for easy management across environments that can be deploying using a single Kustomize command
  - The migration bundle is now configured to automatically support metric monitoring for all Java specification levels

- File Upload
  - File upload has been enhanced to eliminate CORS errors
- Data Collector
  - The Data Collector now uses binaryScanner v23.0.0.2

3.5.2 (May 2023)

- Security Fixes
  - Fix various security issues

3.5.1 (April 2023)

- JBoss
  - Address issue of DTD validation errors when scanning JBoss 6
  - Address issue where outside location scan for JBoss fails
- Security Fixes
  - Fix various security issues

3.5.0 (March 2023)

- Migration Plan
  - Deployment of WebSphere Base in Containers has been streamlined and made consistent with Liberty based deployments through the use of the Runtime Component Operator. It is now possible create cloud based migration plans for Liberty applications that are not yet in containers.
- Recommendations
  - A comparison of each possible migration target is now included in Transformation Advisor
  - Recommendations now include the technology type required for the application, to assist with licensing
- WebSphere ND Clusters
  - The UX has been improved to facilitate the management of large numbers of WebSphere ND Clusters that can appear in Transformation Advisor
- Workspace Management
  - There are significant performance improvements when deleting collection and workspaces
- Cookie Acceptance
  - You will now be prompted to accept essential cookies when authentication is enabled
- Data Collector
  - The Data Collector now uses binaryScanner v23.0.0.1

3.4.2 (February 2023)

- Security Fixes
  - Fix various security issues

3.4.1 (January 2023)

- Security Fixes
  - Fix various security issues

3.4.0 (December 2022)

- Export Report
  - The reports now include the WebSphere Edition required to run the application, showing where right-sizing savings from Base or Core technologies can be achieved with WebSphere Hybrid Edition flexible ratios.
- Recommendations
  - Display of the shared library development effort is now included in the application details page in a manner which is consistent with how common code is displayed
  - Recommendations have been updated to include the latest modernization rules and development effort
- Migration Targets UX
  - The UX has been improved to clearly show the available and selected migration targets
- Multiple Transformation Advisor Instances per Cluster
  - You can now deploy multiple instances of Transformation Advisor to the same OpenShift cluster

3.3.1 (October 2022)

- Fix release
  - Various minor fixes

3.3.0 (September 2022)

- WebSphere ND Clusters
  - The Data Collector will now automatically detect WebSphere ND Clusters and they will appear in the UI
  - The entire WebSphere ND Cluster can be migrated as a single migration plan to deploy all the clustered applications to a single Liberty instance
  - This migration plan will automatically map the WebSphere ND Cluster configuration to an equivalent kubernetes deployment
- Groups
  - An entire Group can now be migrated as a single migration plan to deploy all applications from the Group to a single Liberty instance
- WebSphere Liberty Operator
  - When deploying to the WebSphere Liberty target the migration plan utilised the latest WebSphere Liberty Operator
- Rules
  - The entire ruleset has been overhauled with updated development efforts and severities that reflect the latest best practice for application modernization
- Airgap Install
  - Support for the new ibm-pak plugn which streamlines the deployment of IBM software in an OpenShift disconnected environment

3.2.2 (August 2022)

- Fix release
  - Various minor fixes

3.2.1 (July 2022)

- Fix release
  - Fix air gap installation issue on Red Hat OpenShift
  - Various minor fixes

3.2 (June 2022)

- Groups
  - You can now create any number of Groups of Java applications from your workspace
  - Groups provide a Group estimated total cost allowing you to evaluate a subset of your workspace data
- Bulk Import/Export
  - Bulk Import/Export now supports the import/export of Groups
  - Bulk import now supports a 'noHierarchy' option that will import a flat zip file of Data Collector archive zips
- Data Collector
  - The Data Collector now uses binaryScanner v22.0.0.3 which includes JAX-RPC Conversion Tool Pre-validation and improved performance
  - Transformation Advisor can now accept data produced by the binary scanner's --ta option
- Removal
  - The ability to do migration assessments for MQ QueueManagers has been removed
  - Information on how to migrate IBM MQ can be found here: https://ibm.biz/ta-mod-mq

3.1 (April 2022)

- Reports
  - The exported reports now include a full explanation of how the workspace development costs are calculated, including an exact breakdown for each issue!
- Recommendations
  - The formula to calculate the cost of modernising enterprise entity java beans and for modernising issues across the whole workspace have been adjusted. So you may see the workspace modernisation cost fall!
  - The default target for new recommendations is WebSphere Liberty (instead of Open Liberty)
  - Common code listed for applications is now ordered by complexity, with the most complex common code at the top
- Accessibility
  - There are new layouts for the recommendation and migration plan pages to improve their accessibility
- Data Collector
  - The Data Collector now uses the latest binaryScanner which includes significant performance improvements
- Deprecation
  - The ability to do migration assessments for MQ QueueManagers has been deprecated and will be removed in the next release
- Podman support
  - When running on RHEL you can now use either Docker or Podman
- Security
  - On Openshift, all pods now run inder the default restricted SecurityContectConstraint. You no longer need to create a custom one.

3.0 (January 2022)

Recommendations

- Recommendations are now displayed for the whole workspace along with rollup information
- Common code between all the applications in the workspace is automatically detected in addition to any shared library files that have been defined thus highlighting significant development cost savings by modernizing common code once

Workspace

- You are no longer prompted to create a Collection when you create a workspace. Collection information will be handled

Data Collector

- The Data Collector is now workspace based and no longer tied to a specific collection
- On data upload the host of the scanned machine is detected and automatically used to create the necessary Collections for the data
- The Data Collector now uses binaryScanner v21.0.0.4 which supports moving to Java 17 for Liberty targets

Data Upload

- When uploading data via the UI the Collection for the data is automatically detected by default but can be customized

Installation

- For installation on OCP TA requires two persistence volume claims with ReadWriteOnce access
- One must be at least 20GB in size and the other at least 5GB

Deprecation

- Ability to do migration assessments for MQ Queue Managers has been deprecated and will be removed in a future release
- The recommended procedure for modernizing MQ Queue Managers is in the IBM MQ documentation: https://www.ibm.com/docs/en/ibm-mq/9.2?topic=openshift-migrating-mq-cloud-pak-integration

Removal

- The Business Applications capability has been removed

2.5.0 (September 2021)

```
- Data Collector now uses the binaryScanner v21.0.0.3
- Transformation Advisor now supports user defined rules that are executed during
a scan by the Data Collector
- The results from these rules appear in Transformation Advisor in addition to the
standard rules
- Use the '--no-upload' parameter to prevent the Data Collector output from being
uploaded into the Transformation Advisor server
- The report CSV files that can be exported from Transformation Advisor now
contain additional information regarding the complexity of supporting files, the
reason why applications are considered complex and the ids for workspaces,
collections and profiles to facilitate easier use of the APIs
```

```
- Transformation Advisor now installs via a cluster wide Operator
- When deployed on ROKS, the browser certificates for Transformation Advisor are
now valid by default
```

2.4.4 (June 2021)

```
- Exported reports now include WebSphere Liberty assessment information
- Enhanced capability to rebrand Transformation Advisor
- Recommendations now detect when JAXRPC rules have been mitigated by the axis
framework
- You can now specify a list of applications to exclude from a scan
- Updated configuration now supports very large report files (up to 100MB)
```

2.4.3 (April 2021)

- On install of Transformation Advisor 2.4.3 on Red Hat OpenShift, you must now explicitly select the appropriate license under which you have entitlement.
- Various bug fixes for the Data Collector

2.4.2 (March 2021)

- Transformation Advisor Local is now supported
- Supported on Red Hat Enterprise Linux 7.7+
- Configurable authentication using an external OAuth2 provider
- Additional security hardening options including TLS and HTTPS
- Recommendation generation has a progress indicator
- Landing page can be customized by adding text to the title and/or replacing the TA icon with your own
- Rules for Java SE 14 & 15 migration
- Support for migrating configuration for standalone and Federated LDAP and Application Security Role Bindings for WebSphere
- Support for migrating security configuration specified in tWAS security domains
- File checksum now uses SHA256
- Support for bash 4.3 on z/OS
- Use of an external OAuth2 provider is now supported

2.4.0 (December 2020)

- The landing page has a new streamlined layout
- Workspaces and collections can now be renamed
- The Transformation Advisor support options are now available from all screens in the menu
- The recommendation details page shows estimated development costs at every level of the hierarchy
- Recommendation complexity is now directly linked to the relevant issues that cause it
- Memory requirements are significantly reduced for uploading large datasets
- Upload times are significantly improved for datasets with many dependencies
- Recommendations rendering time has been significantly improved
- Transformation Advisor provides a logout option to invalidate your current session
- Transformation Advisor is now fully accessible
- Re-Certified to meet latest requirements for Enterprise-grade containerized software solutions

2.3.0 (September 2020)

1. Multi-language support
2. More accurate development cost estimate for issues
3. Faster rendering of details pages
4. Option to scan for all targets during data collection
5. All images are now signed
6. Support for upload and display of all SDK 0.5.5 plugin output

Notes (for TA_Local only)

1. Existing TA Local users will need to download new TA Local scripts to run TA 2.3.0. Older scripts will pull the new images and run but uploading your old zips will fail without the new ta_local scripts.
2. Once TA_LOCAL starts you need to use the link to the UI provided (it has an IP in it). You will NOT be able to upload if you use a URL with 'localhost' in it.

2.2.0 (June 2020)

1. Support for deployment of Transformation Advisor on IBM POWER Systems and zLinux
2. Transformation Advisor is now part the Application and Integration CloudPaks (CP4A & CP4I)
3. Improved user experience when viewing the details of an assessment
4. Various bug fixes for the Data Collector

2.1.0 (April 2020)

1. Added analysis and migration plans for both Open Liberty and WebSphere Liberty targets
2. Added migration plan for IBM WebSphere Application Server traditional Base edition in containers
3. Support for modernization assessment tools of IBM Integration Bus
4. Data Collector enhancements
    - Support for full configuration scan for WebSphere v6.1
    - Support for full configuration scan for WebSphere without user credentials
    - Support for scanning for Open Liberty and WebSphere Liberty targets
    - Support for zOS Data Collection

2.0.3 (February 2020)

1. Support for the analysis and migration of applications running on Red Hat JBoss application server
2. Integration with Accelerators for Teams
3. Alignment of assessment and migration targets with OpenShift
4. UI enhancements using Carbon 10 components

2.0.2 (December 2019)

1. Support for Red Hat Openshift Container Platform 4.2
2. User interface uplift to Carbon 10 Design System.
3. Addition of source code flow for application migrations.
4. Some new data collector enhancements, including:

- Support for the collection of applications that uses WebSphere Application Server "Editions"
- Detection of server level shared libraries
- Support for WebSphere variables in shared library paths
- Ability to scope the scan of shared libraries to only those shared libraries used by applications being scanned
- Ability to assess database dependencies for applications (information on database dependencies is currently only available through API's)
- Ability to scan applications that are not managed by the Websphere Application Server deployment manager
- Use of the binaryScanner v19.0.0.4

2.0.1 (September 2019)

1. Red Hat OpenShift Container Platform Support
   IBM Cloud Transformation Advisor now runs on IBM Cloud Pak for Applications on Red Hat OpenShift Container Platform 3.11.
2. Red Hat OpenShift Migration Artifacts
   Creation of migration artifacts that run on IBM Cloud Pak for Applications on Red Hat OpenShift Container Platform.

Application Modernization and how can Transformation Advisor help

Cloud environments and services have exploded the possibilities for deriving value from existing and new data sources. Application requirements in the cognitive era have changed significantly, so in order to respond to this changing market dynamic, businesses must be able to modernize their existing applications while maintaining control of their budgets, managing risk, and investing in new innovations.

What is modernization?

For many, application modernization is just part of a larger digital transformation. The challenge is to find a platform that allows both existing and new applications to converge, providing a unified solution that is robust, secure, scalable, and extensible for existing and new applications alike.

The business expects faster implementation of new ideas to capture new markets, launch new products, and provide better services. For a company to be able to complete its transformation, it will need to understand its existing applications and how each of them can become part of the new era. By understanding the business,

technical, and economic objectives, a company can decide whether an application should be altered or not, and whether it will run best in a private or public cloud environment.

The process of modernizing applications and moving to the cloud can be a large undertaking. Careful planning is needed to prepare business inventory and infrastructure and to determine the best path forward for each application. If you don't yet have a cloud platform but are ready to begin the assessment for your modernization journey, installing IBM Cloud™ Transformation Advisor locally can help you get started. If you want to create a pipeline and deploy to the cloud, Transformation Advisor is included with IBM WebSphere Hybrid Edition.

How does Transformation Advisor help?

IBM Cloud Transformation Advisor has the capability to quickly evaluate your on-premises applications for rapid deployment on WebSphere Application Server and Liberty on public and private cloud environments. The first step is to download and run a custom data collector on your application servers. Results from the scan are uploaded to Transformation Advisor where a detailed analysis is provided.

Transformation Advisor creates a high-level inventory of the content and structure of each application. This information is used to determine complexity and identify the shared library and MQ Queue Manager dependencies for your applications. Transformation Advisor also flags potential issues and estimates a development cost to complete a move to the cloud. Detailed reports with advice, suggestions, and best practices are provided to ensure that the application runs correctly in the preferred cloud environment.

Transformation Advisor helps you on your way to the cloud at these critical stages:

Assessment

- Evaluates applications based on business needs
- Inventories application content and structure
- Provides a migration complexity
- Identifies potential problems for moving to the cloud

Strategic planning

- Identifies key applications that require replacement or upgrade
- Estimates effort for resolving migration issues
- Provides assessments for different modernization options

Execution

- Generates a migration bundle to assist with containerization and deployment
- Creates files tailored to your applications
- Identifies dependencies to help you complete the bundle contents

Transformation Advisor analyzes the following middleware:

Java EE application servers

- WebLogic Server V6.x+
- IBM WebSphere® Application Server V6.1, or later
- JBoss Red Hat® JBoss v4.x+
- Apache Tomcat V6.0 (and later)

- Java applications directly

Messaging

- IBM Integration Bus Version 10.0
- IBM App Connect Enterprise Version 11.0

To get started on your modernization journey, you must understand the approaches that are right for your inventory. Transformation Advisor helps you understand the path that is right for you so you can select the best entry point to deliver value at each step in your journey.
.

IBM Cloud Transformation Advisor considerations for GDPR readiness

This document is intended to help you configure your version of IBM Cloud Transformation Advisor to align with GDPR rules and standards. Here you will find information about features of Transformation Advisor that you are able to edit and adjust, as well as other aspects of the product's use that you should consider to help your organization stay aligned with GDPR rules. This is not an exhaustive list as there are many ways that clients can choose and configure features. There's also a lot of different ways the product can be used in itself as well as with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

Table of Contents

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR

EU GDPR Information Portal:https://gdpr.eu/ IBM GDPR website:http://ibm.com/GDPR

Product Configuration - considerations for GDPR Readiness

The following sections provide considerations for configuring IBM Cloud Transformation Advisor to help your organization with GDPR readiness.

Data Life Cycle

Transformation Advisor is an application that runs on the IBM Cloud Pak for Applications platform. Its purpose is to aid developers in modernizing their application portfolio and migrating their application to cloud platforms. The software gathers and processes meta data from Java Enterprise applications running on IBM WebSphere Application Servers, and creates recommendations on how these applications can be updated or restructured to run in a container on a Cloud platform.

Transformation Advisor deals primarily with technical data that is related to Java programming structures and associated application configuration data. This does not include the gathering or processing of personal data.

Data Collection

Transformation Advisor tool does not collect personal data. It does process technical data such as IP addresses of servers but not personal workstations that could identify an end user. All of this information is only accessible by the system administrator through a management console with role-based access control or by the system administrator through login to an IBM Cloud Pak for Applications platform node.

Another piece of data Transformation Advisor requests from the user is a Git username and password to enable the tool to push the artifacts it generates to help with application migration to the Github repository. This data is not stored and must be re-entered on each use. The data is transmitted to Git and is encrypted in transit.

This is not a definitive list of the types of data that are collected by the Transformation Advisor tool. It is provided as an example for consideration. If you have any questions about the types of data, contact IBM: https://www.ibm.com/data-responsibility/gdpr/.

Data Storage

Transformation Advisor persists technical data in stateful stores on local or remote file systems as files or in a CouchDB database. Consideration must be given to securing all data at rest. Transformation Advisor supports encryption of data at rest in stateful stores that use dm-crypt. For more information, see Encrypting volumes by using dm-crypt (https://www.ibm.com/support/knowledgecenter/SSBS6K_3.2.1/installing/etcd.html?view=kc)

The following items highlight the areas where data is stored, which you might want to consider for GDPR.

Application Data: Transformation Advisor uses CouchDB as a backing data store to persist the technical data that is related to Java programming recommendations and associated application configuration data. CouchDB uses the underlying GlusterFS on which it is deployed for storage. You have to consider encrypting the volumes where GlusterFS storage is deployed for extra security. View IBM Documentation here for more information: (https://www.ibm.com/support/knowledgecenter/SSBS6K_3.2.1/installing/etcd.html?view=kc)

Logging Data: Some technical data such as the IP address of the users browser may be stored in access logs. Logging is configured by default for the IBM Cloud Pak for Applications platform services.

User Authentication Data, (including User IDs and passwords): This type of data is not stored by Transformation Advisor _____

Data Access

Transformation Advisor application data contains no personal information and is accessible through a web user interface. Access to this user interface is authenticated. Transformation Advisor logging data access can be accessed through the Kubernetes kubectl CLI These interfaces are designed to allow administration access and can be secured involving three logical, ordered stages when a request is made: authentication, role-mapping, and authorization.

Data Processing

Users of IBM Cloud Transformation Advisor can control the way that data is processed and secured through system configuration.

Pod security policies are used to set up cluster-level control over what a pod can do or what it can access.

Data-in-transit is protected by using TLS and IPSEC. HTTPS (TLS underlying) is used for secure data transfer between user client and back end services. Users can specify the root certificate to use during installation. All inter-node data traffic can be encrypted out of the box by using IPSEC without changing any applications.

Data-at-rest protection is supported by using dm-crypt to encrypt data.

Data Deletion

Commands, application programming interfaces (APIs), and user interface actions sre provided to delete data that is created or collected by Transformation Advisor.

.

Accessibility

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products.

For information about the commitment that IBM has to accessibility, see the IBM Accessibility Center (www.ibm.com/able).

HTML documentation has accessibility features. PDF documents are supplemental and, as such, include no added accessibility features.

.

Security Bulletins

The security bulletins for Transformation Advisor can be found here

.

Transformation Advisor Deployment Options

Entitlement Options

Get started with assessing your applications. You can try Transformation Advisor at no cost for 90 days on all platforms.

Transformation Advisor does not support running on Kubernetes platforms other than Red Hat OpenShift Container Platform such as IBM Cloud® Kubernetes Service. Note that advice and recommendations by Transformation Advisor is applicable for modernizing Java workload to run on any Kubernetes platform however.

You can continue to use TA beyond the 90-day evaluation period for productive use on certain platforms if you have entitlement to any of the IBM products in the license information.

Platform/OS	Support	90 Day Free Evaluation	Productive Use	Installation	Hardware
OCP 4.12,4.14, 4.15	Full	Yes	Yes apply key	install via Operator	x86_64/Power/IBM Z
RHEL 7.9+	Full	Yes	Yes apply key	installation guide	amd64
Windows v10, v11	Limited	Yes	No	installation guide	Desktop/Laptop

SUPPORT TYPE: Transformation Advisor is fully supported for installation on Red Hat Enterprise Linux. Limited Support is available for installation on Windows and is provided on a best-efforts basis. Security features such as authentication, TLS and proxy configuration are not supported on Windows.

Moving from a desktop evaluation install to a supported install

If you have been using the free evaluation on your desktop/laptop and are moving to running TA on OCP or RHEL you can keep your data. If you kept your Data Collector .zip files, you can upload your data to the cloud instance again. Alternatively, you can export the data from Transformation Advisor on your desktop and import to your supported instance by following the export and import instructions.

.

Planning

Please read the following documents carefully in advance of installing Transformation Advisor on Red Hat OpenShift Container Platform.

- OpenShift Container Platform compatibility
- Sizing
- Storage Considerations
- Licensing
- Security Considerations
- Image Registry Access

.

IBM Cloud Transformation Advisor and Red Hat OpenShift Container Platform compatibility

The IBM Cloud Transformation Advisor Operator is supported on the following versions of Red Hat® OpenShift® Container Platform (OCP):

- OCP 4.12, 4.13, 4.14

Running on any of the following hardware:

- Linux on x86_64
- Linux on Power (ppc64le)
- Linux on IBM Z

Note

- IBM Cloud Transformation Advisor is not supported on Power with FIPS enabled

.

Sizing

System requirements can vary depending on the size and purpose of your deployment.

Minimum OpenShift Cluster Sizing

Typically, Transformation Advisor will be one of many applications running on the OpenShift cluster and that needs to be taken into account when sizing your OpenShift cluster. The minimum recommended OpenShift requirements for running Transformation Advisor are outlined in the following table:

Mstr/Worker	Nodes	CPU (Cores/Node)	Memory (GB/Node)
Mstr	3	8	16
Worker	3	8	16

Cluster Resources Required by Transformation Advisor

This section describes the resources required to install and run Transformation Advisor. Choose a configuration profile that aligns with your usage, and allocate resources to Transformation Advisor

accordingly. Please note that Transformation Advisor *does not* support a High Availability configuration.

Configuration Profile Definitions

Configuration Profile Name	Usage
Starter	minimum size for functioning product, ie demo use case
Production	recommended size for production environment

Starter Profile Configuration

Subsystem	CPU	Memory (GB)	Disk Space (GB)
CouchDB	0.5	1	20
GraphDB	0.5	1	5
Server	0.5	1	N/A
UI	0.5	1	N/A
Operator	0.4	0.5	N/A

Production Profile Configuration

Subsystem	CPU	Memory (GB)	Disk Space (GB)
CouchDB	1.5	3	20
GraphDB	1.5	3	5
Server	1.5	3	N/A
UI	1.5	3	N/A
Operator	0.5	0.5	N/A

These values can be modified in the custom resource yaml prior to install. See Configuring for more details.

.

Storage Considerations

The following section is a guide to how you can configure persistence for Transformation Advisor. It is not a thorough guide to storage on a Kubernetes cluster in general. Please refer to the Kubernetes documentation if you are not familiar with concepts mentioned in this section. Refer to Red Hat OpenShift documentation and your storage administrator to understand the persistence options available on your specific version of OpenShift.

Transformation Advisor requires persistence to be configured in order to persist the database data. When Transformation Advisor starts for the first time, it will create a **`PersistentVolumeClaim`** (unless you have set the option to use an existing claim). The persistence options provided by Transformation Advisor, allow you to control the attributes of the **`PersistentVolumeClaim`** that Transformation Advisor creates. The **`PersistentVolumeClaim`** that Transformation Advisor creates, must bind to a **`PersistentVolume`** on the cluster. The properties set for the **`PersistentVolumeClaim`** must be compatible with the **`PersistentVolumes`** available on your cluster.

From Transformation Advisor 3.0.x onwards, **two** **`PersistentVolumeClaims`** are required.

The persistence may be configured from the UI form view or the YAML view when installing the Transformation Advisor instance.

It is **NOT** recommended to use local volumes as backing for persistence for Transformation Advisor.

Supported Storage Providers

The following storage providers are supported:

- OpenShift Container Storage / OpenShift Data Foundation version 4.x, from version 4.2 or higher
- IBM Cloud Block storage and IBM Cloud File storage
- Portworx Storage, version 2.5.5 or above
- File storage from IBM Spectrum Fusion/Scale
- Amazon Elastic File Storage (EFS) for RWX mode access

Transformation Advisor Persistence Options

The following properties can be configured to control the persistence that Transformation Advisor uses. Set the options according to your requirements and constraints when installing the Transformation Advisor instance. See Example Persistence Configurations to see how to use these configuration options.

Enable / Disable

Configuration setting: **`persistence.enabled: <true|false>`**

Although it is NOT recommended for a production install, you can also turn the persistence off. Transformation Advisor will have full functionality with persistence disabled except **importantly, if persistence is disabled, you will lose all of your Transformation Advisor data when the database container restarts**.

Access modes

The access mode determines the **`PersistentVolume`** that the Transformation Advisor **`PersistentVolumeClaim`** can bind to. The value for access mode must be supported by the type of storage that is available on your cluster. For example, if you are installing Transformation Advisor on public cloud, the accessMode is **`ReadWriteOnce (RWO)`**. Consult your storage administrator for more details.

Configuration setting: **`persistence.accessMode: <"ReadWriteOnce">`**

Transformation Advisor supports **`ReadWriteOnce (RWO)`** access mode. **`ReadWriteMany (RWX)`** and **`ReadOnlyMany (ROX)`** access modes are *not* supported by Transformation Advisor.

Storage class

Configuration setting: **`persistence.storageClassName: <"storage class name">`**

A **`StorageClass`** provides a way for administrators to describe the "classes" of storage they offer. If the **`PersistentVolumes`** configured for your cluster have a storage class, then you can set the **`storageClassName`** property in Transformation Advisor, so that **`PersistentVolumeClaim`** that Transformation Advisor creates will bind to those **`PersistentVolumes`**.

If you are configuring persistence in the UI form view, the Storage Class widget will be pre-populated with the StorageClasses available on your cluster. Alternatively, you can navigate to storage classes in the navigation in OpenShift UI to see what storage class are available on your cluster.

Alternatively, you can use the following command to see the storage classes available.

```
oc get storageclass
```

Storage type and permissions

If you are installing on an environment that uses block storage - (specified using a **storageClass**), Transformation Advisor can bind to the storage and set the correct permissions for that storage. If you are installing on an environment that uses file system storage such as NFS, you must ensure that Transformation Advisor has read/write permissions to the storage on the host. For details, see Setting permissions for NFS.

If there are not sufficient permissions for the storage that Transformation Advisor uses, the CouchDB pod fails to start and a permission denied message is added to the couch db pod logs:

```
[error] 2019-08-23T21:08:33.974674Z nonode@nohost <0.256.0>
-------- Could not open file ./data/_users.couch: permission denied
```

Setting permissions for NFS

Note: From Transformation Advisor 3.0.x onwards, two **PersistentVolumeClaims** are required, and you need to repeat the following procedures for both **PersistentVolmueClaims** accordingly. When shared storage is mounted into a container, it is mounted with the same ownership and permissions that are found on the exported NFS directory. The container that uses the storage is not run with that owner and might not have permission to read or write to the storage. To enable read/write permission to the storage, consider the following options:

Option A: Control permissions with the supplementalGroups setting

Configuration setting: **persistence.supplementalGroups: <[id1,id2...]>**

Supplemental groups are regular Linux groups and are typically used for controlling access to shared storage, such as NFS. When a process runs in Linux, it has a UID, a GID, and one or more supplemental groups. The supplemental groups can be set for a container's main process. The supplemental group gives the container process group ownership on the storage and consequently the group level permissions to allow the container process to read/write to the storage.

Consider the following example where the underlying host storage is located at **/netstore/transadv/**. The nfsnobody group might have full permissions on the shared directory when NFS is shared with the root_squash option:

```
> ls -ld /netstore/transadv/
drwxrwx---. 7 nfsnobody nfsnobody 182 Aug 31 22:30 /netstore/transadv/
```

To control permissions with the **supplementalGroups** setting, and give the container the ability read/write to the storage, add the **nfsnobody** GID to the supplemental groups as follows:

1. Get the nfsnobody GID using the **id** command.

```
> id nfsnobody
uid=65534(nfsnobody) gid=65534(nfsnobody) groups=65534(nfsnobody)
```

2. Add the GID (65534 in this case) using the **supplementalGroups** setting, e.g. configuration.

```
persistence:
...
<couchdb | neo4j>
supplementalGroups: [65534]
```

Option B: Create a PersistentVolume specifically for use with Transformation Advisor

Configuration setting: **`persistence.existingClaim: <"claim name">`**

Creating a **`PersistentVolume`** specifically for Transformation Advisor allows you to set the access permissions for storage usedby that **`PersistentVolume`**.

This option describes how to: 1. Create a NFS **`PersistentVolume`** specifically for Transformation Advisor 2. Create a **`PersistentVolumeClaim`** to bind to that **`PersistentVolume`** 3. Use that claim (with appropriate permissions) for Transformation Advisor

We will manually create a **`PersistentVolumeClaim`** and ensure that the storage backing its **`PersistentVolume`** has the correct (open) permissions. You can ensure a **`PersistentVolumeClaim`** binds to a specific **`PersistentVolume`** with selector and label attributes.

1. Define an NFS PersistentVolume. The following sample is an NFS **`PersistentVolume`** definition. Substitute the path and server values for your own environment.

apiVersion: v1 kind: PersistentVolume metadata: name: tanfspv labels: pvc_for_app: "ta" spec: capacity: storage: 20Gi accessModes:

- ReadWriteOnce nfs: path: /nfs/ta_storage server: xxx.xxx.xxx.xxx persistentVolumeReclaimPolicy: Recycle

  `**Make sure that the directory at the path has open permissions so that the container can read/write there:**`

chmod -R 777 /netstore/transadvopen/

`1. Define a PersistentVolumeClaim. The following sample is a PersistentVolumeClaim definition that binds to the `PersistentVolume` described in the previous sample. Ensure that you create the `PersistentVolumeClaim` in the Transformation Advisor namespace.`

apiVersion: v1 kind: PersistentVolumeClaim metadata: name: tapvc spec: accessModes:

- ReadWriteOnce resources: requests: storage: 20Gi selector: matchLabels: pvc_for_app: "ta"

If you want Transformation Advisor to use a specific PersistentVolumeClaim, add the name of the claim using the **`existingClaim`** property.

Option C: Update permissions for the PersistenceVolume after Transformation Advisor has tried and failed to start

If you are using NFS storage, and you do not complete option A or B, when Transformation Advisor attempts to start - you will see that the Couch DB pod will not start. If you look at the logs for the Couch DB pod you will see a permissions issue:

```
[error] 2019-08-23T21:08:33.974674Z nonode@nohost <0.256.0>
-------- Could not open file ./data/_users.couch: permission denied
```

At this point, you can find the **`PersistentVolumeClaim`** that Transformation Advisor has created:

```
oc get pvc
```

That will reveal the **`PersistentVolume`** that it has bound to (under the **`VOLUME`** header).

You can describe that **`PersistentVolume`** to discover the path for the storage:

```
oc describe pv tanfspv | grep Path:
```

You can then open the permissions on that path **`chmod -R 777 <path>`**

The Couch DB should then restart successfully within a few minutes. Or, you can delete the Couch DB pod and it will be automatically restarted.

Storage Classes Red Hat OpenShift Container Platform on IBM Cloud

IBM Cloud provides a number of storage classes for OpenShift clusters. Some of the storage classes use block storage and can be identified by the "-block-" in their name. Other storage classes use file storage and can be identified by the "-file-" in their name. The storage class with the name "default" uses file storage also. Please note that the storage with the name "default" may be different to the storage class that is marked as the default storage class for the cluster.

Transformation Advisor supports the use of any of the block based storage classes. Transformation advisor does not support the use of the file based storage (including the storage class called "default"), except for the storage classes that have a "-gid" suffix, i.e.

- ibmc-file-bronze-gid
- ibmc-file-silver-gid
- ibmc-file-gold-gid

These storage classes allow you to specify a supplemental group when installing so that the Transformation Advisor non-root containers can write to the storage. When using these **`*-gid`** storage classes, please specify the supplementalGroups of 65531 at install time.

Please refer to the following document for more information on storage classes on Red Hat OpenShift on IBM Cloud: Storage class reference.

Example Persistence Configurations

Please refer to the Installing document on how to apply these example configurations.

EXAMPLE 1: Using storageClass

Use the storage class to create a **`PersistentVolumeClaim`** that will bind to a **`PersistentVolumeClaim`** of that class. In this case, we are using "ibmc-block-silver" on public cloud.

For a UI install, the custom resource YAML would look like this:

```
...
  persistence:
    enabled: true
    <couchdb | neo4j>
      accessMode: "ReadWriteOnce"
      size: 20Gi
      useDynamicProvisioning: true
      existingClaim: ""
      storageClassName: "ibmc-block-silver"
```

```
      supplementalGroups: []
...
```

For a CASE install the arguments would look like this:

```
--args "... --storageClass ibmc-block-silver --accessMode ReadWriteOnce ..."
```

EXAMPLE 2: Using supplementalGroups

In the following example, we are using the **`supplementalGoups`** setting to ensure the database pods have sufficient permissions on NFS storage we are using. The group ownership of the shared file system has a GID of **`65534`**

For a UI install, the custom resource YAML would look like this:

```
...
  persistence:
    enabled: true
    <couchdb | neo4j>
      accessMode: "ReadWriteOnce"
      size: 20Gi
      useDynamicProvisioning: true
      existingClaim: ""
      storageClassName: ""
      supplementalGroups: [65534]
...
```

For a CASE install the arguments would look like this:

```
--args "... --supplementalGroups [65534] ..."
```

EXAMPLE 3: Using existingClaim

In the following example, a persistence volume claim called **`my-ta-pvc`** has been already set up by the user. The persistence volume claim has open permissions to allow the database container to write to it.

For a UI install, the custom resource YAML would look like this:

```
...
  persistence:
    enabled: true
    <couchdb | neo4j>
      accessMode: "ReadWriteOnce"
      size: 20Gi
      useDynamicProvisioning: true
      existingClaim: "my-ta-pvc"
      storageClassName: ""
      supplementalGroups: []
...
```

For a CASE install the arguments would look like this:

```
--args "... --persistenceClaimCouchDB my-ta-pvc --persistenceClaimNeo4j my-ta-
pvc2..."
```

Deleting a Persistence Volume Claim:

NOTE: Do not delete `PersistentVolumes` or `PersistentVolumeClaims` unless you are entirely satisfied that they are not being used by containers. Deleting `PersistentVolumes` or

`PersistentVolumeClaims` may interfere with running applications and/or delete data that cannot be recovered.

In a normal uninstallation of Transformation Advisor, the **`PersistentVolumeClaim`** that Transformation Advisor has created will be deleted, and the **`PersistentVolume`** that the claim binds to will be released for use.

If you need to manually delete **`PersistentVolumes`** or **`PersistentVolumeClaims`** please consult the relevant Kubernetes documentation

.

Licenses under which you have entitlement to use Transformation Advisor

Product name & version	License ID	Link to License
IBM WebSphere Hybrid Edition 5.1	L-ZZBG-6V3K4K	Review License
IBM Cloud Transformation Advisor 3.10.0 (Evaluation)	L-APJC-75QQ5K	Review License
IBM Cloud Pak for Integration 2023.2.1	L-YBXJ-ADJNSM	Review License
IBM Cloud Pak for Integration Limited Edition 2023.2.1	L-PYRA-849GYQ	Review License
IBM Cloud Pak for Applications Advanced 5.3	L-REBM-59QT8V	Review License
IBM Cloud Pak for Applications Standard 5.3	L-ZUVT-EDS78A	Review License
IBM Cloud Pak for Applications Limited 5.3	L-KULQ-TG6TN5	Review License
IBM WebSphere Application Server for z/OS 8.5.5	L-CTUR-C7K3YZ	Review License
IBM WebSphere Application Server for z/OS 9.0.5	L-CTUR-CBPUER	Review License
IBM WebSphere Automation v1.7.2	L-UMDR-KJVYBM	Review License

License Usage on Red Hat OpenShift Cluster

The following bash script can be used to retrieve license information for all the instances of Transformation Advisor on a cluster. The script can be run on RHEL or on MacOS. You must be **`oc logged in`** to the cluser as an admin to run the script.

```
#!/bin/bash

# ---------------------------------------------
# Licensed Materials - Property of IBM
# (C) Copyright IBM Corporation 2023
# ---------------------------------------------

#
# GLOBALS
#
report_name="./ibm-transformation-advisor-license-report.txt"

#
# FUNCTIONS
#

#
# Find the date 180 days in past
```

```
#
function get_180_days_ago() {
        local time_in_past_180=""

        # Check if the date command has the -j option
        if date -j >/dev/null 2>&1; then
                # Use the -j option for macOS
                time_in_past_180=$(date -j -v-180d +'%Y-%m-%d')
        else
                # Use the -d option for Linux
                time_in_past_180=$(date -d "-180 days" +%Y-%m-%d)
        fi

        # Print the Unix timestamp
        echo "${time_in_past_180}"
}


#
# MAIN ENTRY
#

time_in_past_180=$(get_180_days_ago)

# Check logged into OCP cluster
if oc whoami >/dev/null 2>&1; then
        current_user=$(oc whoami)
        current_ocp_cluster=$(oc whoami --show-server)
else
        echo "ERROR: You are not logged into an OpenShift cluster."
        echo "Please log into an OpenShift cluster as an admin user to run this
report"
        exit 1
fi

# Check if report file already exists
if [[ -f ${report_name} ]]; then
        read -r -p "A file called ${report_name} already exists. Overwrite?
[yes/no] " response
        case "$response" in
                [yY][eE][sS])
                echo "OK. Overwriting ${report_name}"
                rm -f ${report_name}
                ;;
                *)
                echo "Move, remove or rename ${report_name} before running this
tool."
                exit 1
                ;;
        esac
fi

echo "Generating report..."
echo -n "."


# Write report header
echo
"##############################################################################"
> $report_name
echo "# IBM Transformation Advisor License Report" >> $report_name
echo "#" >> $report_name
echo "# Time: $(date)" >> $report_name
echo "#" >> $report_name
```

```
echo
"##############################################################################"
>> $report_name

# Write report summary
echo "" >> $report_name
echo
"******************************************************************************"
>> $report_name
echo "SUMMARY" >> $report_name
echo "" >> $report_name
echo "User: $current_user" >> $report_name
echo "OpenShift endpoint: $current_ocp_cluster" >> $report_name

num_instances=$(oc get transadvs.ta.ibm.com --all-namespaces --no-headers | wc -l
| xargs)

if [[ ${num_instances} = 0 ]]; then
        echo "No instances of IBM Transformation Advisor were found on the
cluster. All namespaces were checked" >> $report_name
        exit 0
else
        echo "Number of IBM Transformation Advisor instances installed on the
cluster: ${num_instances}" >> $report_name
fi
echo -n "."

echo "" >> $report_name
echo
"******************************************************************************"
>> $report_name
echo "DETAILS" >> $report_name
echo "" >> $report_name

# Iterate over all the instance on the cluster
oc get transadvs.ta.ibm.com --all-namespaces -o custom-
columns=NAMESPACE:.metadata.namespace --no-headers | while read -r inst_namespace;
do

        echo -n "."

        num_instances=$((num_instances+1))

        namespace=${inst_namespace}
        name=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o custom-
columns=NAME:.metadata.name --no-headers)
        creation_timestamp=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o
custom-columns=CREATION:.metadata.creationTimestamp --no-headers)
        license=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o custom-
columns=LICENSE:.spec.license.aLicenseType --no-headers)

        echo "----------------------------------------" >> $report_name
        echo "Namespace: ${namespace}" >> $report_name
        echo "Instance Name: ${name}" >> $report_name
        echo "Created: ${creation_timestamp}" >> $report_name
        echo "License: ${license}" >> $report_name

        if [[ ${license} = *"Evaluation"* ]]; then
                creation_time=$(echo ${creation_timestamp} | sed "s/T.*//g")
                if [[ "${creation_time}" <  "${time_in_past_180}" ]]; then
                        echo "" >> $report_name
                        echo "WARNING: This instance was created more than 180
days ago and is using an evaluation license." >> $report_name
                        echo "Please upgrade to a full license." >> $report_name
```

```
                fi
        fi

        echo "" >> $report_name

done

echo "" >> $report_name
echo
"*******************************************************************************"
>> $report_name

echo ""
cat ${report_name}
echo "Report complete. Saved as ${report_name}"
echo ""
```

.

Security Considerations

The following sections describe the key considerations to successfully secure your deployments on Red Hat OpenShift.

- Verifying Signatures of TA Artifacts
- User management
- Routes
- Network Policies
- Secrets

Verifying Signatures of TA Artifacts

The public keys, certificates and certificate chains that are required for verifying the TA signatures are available for download here: taPublicKeys.zip

Unzip the archive to access the following files:

- `PRD0004063key.pem.cer`
- `PRD0004063key.pem.chain`
- `PRD0004063key.pem.pub.key`
- `PRD0004063key.pub.asc`

Subsequent sections describe how to use these files to verify the TA artifact signatures.

Verifying TA Image Signatures

The Transformation Advisor AMD64 images are signed using GPG simple signing. The signature can be verified by the **`skopeo`** OR **`podman`** tools as follows:

1. Create a **`policy.json`** file that configures the verification. For example:

```
{
   "default":  [{"type": "reject"}],
   "transports": {
```

```
        "docker": {
            "cp.icr.io/cp/icpa": [{
                "type": "signedBy",
                "keyType": "GPGKeys",
                "keyPath": "<KEY_LOCATION>/PRD0004063key.pub.asc"
            }],
            "icr.io/cpopen": [{
                "type": "signedBy",
                "keyType": "GPGKeys",
                "keyPath": "<KEY_LOCATION>/PRD0004063key.pub.asc"
            }],
            "icr.io/appcafe": [{
                "type": "signedBy",
                "keyType": "GPGKeys",
                "keyPath": "<KEY_LOCATION>/PRD0004063key.pub.asc"
            }]
        }
    }
}
```

2. Method 1: Verify using `skopeo copy...`

```
skopeo copy --policy ~/policy.json docker://cp.icr.io/cp/icpa/transformation-
advisor-ui@sha256:1234... dir:./image-dir --src-creds iamapikey:myapikey
```

3. Method 2: Verify using `podman pull...`

```
podman pull --signature-policy ~/policy.json cp.icr.io/cp/icpa/transformation-
advisor-ui@sha256:1234..
```

Please see Red Hat OpenShift Container Platform docs for information configuring your OpenShift cluster to verify the image signatures.

Verifying TA Local Zip Signature

For more information on TA Local, and how to download the zip file, see Installing IBM Cloud Transformation Advisor on RHEL. When downloading the zip, you should also download the **`<filename>.zip.cosign.sig`** file to allow you to verify the integrity of the zip file.

There are three ways to verify the signature, according to preference. The public keys, certs and chains needed for the following steps can be downloaded from the links at the start of this section.

Method 1: Using the PEM encoded public key

Prerequisites:

- cosign utility. To install see here: https://github.com/sigstore/cosign/releases
- The PEM public key: `PRD0004063key.pem.pub.key`
- The signature file: `transformationAdvisor-${VERSION}.zip.cosign.sig`
- The zip artifact: `transformationAdvisor-${VERSION}.zip`

```
cosign verify-blob --key PRD0004063key.pem.pub.key --signature
transformationAdvisor-${VERSION}.zip.cosign.sig
transformationAdvisor-${VERSION}.zip
```

Method 2: Using the PEM encoded public certificate

Prerequisites:

- cosign utility. (To install see here: https://github.com/sigstore/cosign/releases). It is recommended to use version 1.x of `cosign`. If using version 2+, then you must add the flag `--insecure-ignore-tlog=true` when verifying.
- The PEM public certificate: PRD0004063key.pem.cer
- The certificate chain: PRD0004063key.pem.chain
- The signature file: transformationAdvisor-${VERSION}.zip.cosign.sig
- The zip artifact: transformationAdvisor-${VERSION}.zip

```
cosign verify-blob --cert PRD0004063key.pem.cer --cert-chain
PRD0004063key.pem.chain --signature
transformationAdvisor-${VERSION}.zip.cosign.sig
transformationAdvisor-${VERSION}.zip
```

Method 3: Using the PEM encoded public key with openssl

Prerequisites:

- `openssl`
- The PEM public key: `PRD0004063key.pem.pub.key`
- The signature file: `transformationAdvisor-${VERSION}.zip.cosign.sig`
- The zip artifact: `transformationAdvisor-${VERSION}.zip`

```
openssl enc -d -A -base64 -in transformationAdvisor-${VERSION}.zip.cosign.sig -out
./transformationAdvisor-${VERSION}.zip.bytes.sig
openssl dgst -verify PRD0004063key.pem.pub.key -keyform PEM -sha256 -signature
./transformationAdvisor-${VERSION}.zip.bytes.sig -binary
transformationAdvisor-${VERSION}.zip
```

User Management

By default, the Transformation Advisor instance is automatically configured to use the OpenShift Container Platform OAuthClient. You can configure Transformation Advisor to use a third-party authentication source.

For more information on how to configure a third-party authentication source, see Configuring IBM Cloud Transformation Advisor.

Transformation Advisor does not differentiate roles. All authenticated users have access to all actions in the product.

Routes

Routes are used to provide external access to cluster resources. Transformation Advisor creates three dynamic routes based on the project that it is installed in and the cluster domain. The routes are as follows:

Route	Description	Example URL
`<ta instance>`-openapi-route	Access the API via Swagger UI	https://openapi.myproj-ta.apps.kgta.cp.fyre.ibm.com/openapi/ui
`<ta instance>`-server-route	Access the liberty server. Not to be used directly by end users.	https://myproj-ta.apps.kgta.cp.fyre.ibm.com/lands_advisor
`<ta instance>`-ui-route	Access the UI. Main entry point for end users.	https://openshift-operators-ta.apps.kgta.cp.fyre.ibm.com

Internal TLS Certificates

You can provide custom certificates for internal TLS communications between the Transformation Advisor pods.

For more information, see Enable Bring Your Own Key (BYOK).

Network Policies

Network policies control the ingress and egress traffic to and from the Transformation Advisor pods.

Egress Network Policy

If your cluster uses the OpenShift SDN default Container Network Interface (CNI) network provider, Transformation Advisor automatically configures an **`EgressNetworkPolicy`** to limit egress traffic from the Transformation Advisor pods. The default configuration is suitable for most cases, but you can customize the allowed egress or disable it altogether.

For more information, see Configuring IBM Cloud Transformation Advisor.

Ingress Network Policy

Transformation Advisor automatically configures ingress network policies to restrict incoming traffic to only essential traffic.

For more information, see Network policy.

Secrets

Transformation Advisor securely stores the necessary credentials for its operation by using a **`Secret`**. Transformation Advisor automatically generates the **`Secret`** during installation with unique and random values. Users can also provide their own **`Secret`**, partially or fully, for the necessary values. Transformation Advisor automatically generates any values that are not supplied by the user.

For instance, users need to provide the OAuth client ID and secret on setting up third-party authentication. For more information, see Configuring IBM Cloud Transformation Advisor.

Image Registry Access

The images that are required for Transformation Advisor are hosted in the following locations.

Registry	Description
`icr.io`	OLM Images (operator, bundle, and catalog) are located in **`icr.io/cpopen`**. No authentication is required for access. Trial product images are stored in **`icr.io/appcafe`**. No authentication is required for access.
`cp.icr.io`	Entitled product images are located in **`cp.icr.io/cp/icpa`**. An entitlement key is required to access.

Your cluster needs access to these registries. If you operate block or allow lists for registries in your cluster, then you need to update the **`image.config.openshift.io/cluster`** resource to allow access to **`icr.io`** and **`cp.icr.io`**.

Trial versus Entitlement Image Access

Transformation Advisor can be installed with a time limited evaluation license (fully featured), or as an entitled product. For more information, see Licenses under which you have entitlement to use Transformation Advisor.

- If installing as an evaluation, you can ignore the following instructions that describe how to get access to the entitled registry. In this case, the required images will be pulled from a public registry (**`icr.io/appcafe`**) and no access details are required.
- If installing Transformation as an entitled product, you will need to have an entitled registry key.

Entitlement to Transformation Advisor is obtained by purchasing one of a number of other IBM products. To find your entitlement key you must first ensure that you have entitlement to any one of the IBM products listed here. Then log in to MyIBM Container Software Library with the IBMid and password that are associated with the entitled software. Your Entitlement Key will be displayed there.

See #entitlement-key for more on creating the **`ibm-entitlement-key`** secret.

Migration to IBM Cloud Container Registry

As of Transformation Advisor *2.4.3*, some of the Transformation Advisor images (the operator, bundle, and catalog) were migrated from Docker Hub (**`docker.io/ibmcom`**) to the IBM Cloud Container Registry (**`icr.io/cpopen`**).

This change affects all of your IBM Operator Catalog provided software, not just Transformation Advisor. For more information, see Migrating from Docker to the IBM Container Registry.

For Transformation Advisor, the following are the changes that are required:

If you operate in an internet-connected Red Hat® OpenShift® Container Platform cluster:

- Update the IBM Operator Catalog to point to **`icr.io/cpopen`**, instead of **`docker.io/ibmcom`**. For more information, see Migrating from Docker to the IBM Container Registry.
- If you are upgrading Transformation Advisor from a version *BEFORE 2.4.3*, apply an **`ImageContentSourcePolicy`**. For more information, see Migrating from Docker to the IBM Container Registry.

If you operate in a disconnected (air-gapped) Red Hat® OpenShift® Container Platform cluster:

- When mirroring the Transformation Advisor images, update the manifest of images to replace **`docker.io/ibmcom`** with **`icr.io/cpopen`**. For more information, see Installing IBM Cloud Transformation Advisor in a Disconnected/Air-gapped Environment.

Software Product Compatibility Reports (SPCR)

When you plan an installation, it is important to review the list of supported operating systems, detailed system requirements, and hardware requirements. To get this information, you can view a software product compatibility report.

1. Go to the Software Product Compatibility Reports page.
2. Click **Create a report** for the type of report you want to create.
3. For the **Full or partial product name** field, enter `Transformation Advisor` and click the **Search product** icon.
4. In the search results, select a version in the **Version list** and click **Submit**.

Learn the steps to quickly and easily generate custom IBM software product compatibility reports.

Installing IBM Cloud Transformation Advisor

IBM Transformation Advisor can be installed from IBM Operator Catalog in the OperatorHub in the Red Hat OpenShift UI.

For a disconnected or air-gap install, Transformation Advisor can be installed using the ibm-pak-plugin and the CASE package. See airgap install for more details.

As part of the installation procedure for Transformation Advisor (for both UI install and air-gap install), cluster administration tasks are required to be performed by a cluster administrator. See Cluster Admin Tasks for more details.

1. Install Red Hat OpenShift Container Platform
2. Configure Storage
3. Add the IBM Operator Catalog
4. Choose Installation Mode
5. Perform Cluster Administration Tasks for Operator Installation
6. Install Operator
7. Create entitlement key
8. Perform Cluster Administration Tasks for Instance Installation
9. Create Transformation Advisor instance

Uninstall Transformation Advisor

- Uninstalling Transformation Advisor

1. Install Red Hat OpenShift Container Platform

Please see [Installing Red Hat OpenShift Container Platform] (https://www.ibm.com/docs/en/cloud-paks/1.0?topic=installing-openshift-container-platform-clusters)

2. Configure Storage

It is highly recommended to configure storage for Transformation Advisor. You may explore the product functionality without storage configured (and setting persistence enabled to false), however you will lose your

data when the Transformation Advisor database container restarts. For a full description of storage considerations, please see Storage Considerations.

3. Add the IBM Operator Catalog

Note: Transformation Advisor 2.3.x was delivered in the Red Hat Certified catalog **and has now been removed from that catalog**. Transformation Advisor 2.4+ is only be available in the IBM Operator Catalog. See Upgrading for more on how to upgrade from 2.3 to 2.4.

Please follow the Adding the IBM operator catalog documentation.

4. Choose Installation Mode

Transformation Advisor supports two installation modes:

- All namespaces on the cluster (default)

  Operator will be available in all Namespaces.

- A specific namespace on the cluster

  Operator will be available in a single Namespace only.

Depending on the installation mode chosen, you will need to perform different installation tasks as a cluster administrator. See Cluster Admin Tasks.

5. Perform Cluster Administration Tasks

As part of the installation, the cluster administrator will be required to perform some tasks. Depending on your installation mode, you will need to complete different cluster administration tasks, see subsequent sections for details. For more information on configuring operators with `OperatorGroups` please see: https://olm.operatorframework.io/docs/advanced-tasks/operator-scoping-with-operatorgroups.

Cluster Administration Tasks for an All Namespaces Operator Installation

The cluster administrator must perform the operator installation for both the Red Hat OpenShift UI installation and the CASE installation.

Cluster Administration Tasks for a Single Namespace Operator Installation

The cluster administrator must perform the following tasks:

1. Create a project for the operator and instance.

   This can be done from the OpenShift UI as follows:

   a. Click the hamburger icon of the Red Hat OpenShift Container Platform dashboard.
   b. Expand the **Home** menu and select **Projects**.
   c. Click **Create Project**.
   d. Name the project and click **Create**.

   Alternatively, you can use the `oc new-project NAMESPACE_PLACEHOLDER` command from the command line, where NAMESPACE_PLACEHOLDER is substituted for the actual name of the project you want to create.

2. Set up Role Based Access Control (RBAC) for Operator.

   The YAML in this section creates a **`ServiceAccount`**, **`OperatorGroup`**, **`Role`**, **`RoleBinding`**, **`ClusterRole`** and **`ClusterRoleBinding`** resource which together configure RBAC for the operator. Apply the YAML to your cluster:

a. Copy the YAML and replace all occurrences of NAMESPACE_PLACEHOLDER with the project that you are installing into.
b. Click the **+** button in the OpenShift UI and paste the YAML into the Import YAML window.
c. Click **Create**.

Alternatively, you can save the YAML to file, replace NAMESPACE_PLACEHOLDER with the project you are installing to, and create the resources using the **`oc apply -f <filename>`** command.

```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: ta-operator-sa
  namespace: NAMESPACE_PLACEHOLDER

---

apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: ta-ownnamespace-operator-group
  namespace: NAMESPACE_PLACEHOLDER
spec:
  targetNamespaces:
    - NAMESPACE_PLACEHOLDER

---

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: ibm-transformation-advisor
  namespace: NAMESPACE_PLACEHOLDER
rules:
  - apiGroups: ["networking.k8s.io"]
    resources: ["networkpolicies"]
    verbs: ["get", "list", "watch", "create", "delete", "patch"]
  - apiGroups: ["config.openshift.io"]
    resources: ["ingresses", "infrastructures", "dnses"]
    verbs: ["get"]
  - apiGroups: ["oauth.openshift.io"]
    resources: ["oauthclients", "oauthclients/finalizers"]
    verbs: ["get", "list", "create", "delete"]
  - apiGroups: ["operators.coreos.com"]
    resources: ["clusterserviceversions"]
    verbs: ["get", "list", "watch", "create", "delete", "patch"]
  - apiGroups: ["apiextensions.k8s.io"]
    resources: ["customresourcedefinitions",
"customresourcedefinitions/finalizers"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: ["batch"]
    resources: ["jobs"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: [""]
```

```
    resources: ["configmaps", "persistentvolumeclaims", "pods", "services",
"secrets", "serviceaccounts", "events"]
    verbs: ["*"]
  - apiGroups: ["apps"]
    resources: ["deployments", "statefulsets"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: ["monitoring.coreos.com"]
    resources: ["servicemonitors"]
    verbs: ["get", "create"]
  - apiGroups: ["apps"]
    resourceNames: ["ta-operator"]
    resources: ["deployments/finalizers", "statefulsets/finalizers"]
    verbs: ["update"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get"]
  - apiGroups: ["apps"]
    resources: ["replicasets"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: ["ta.ibm.com"]
    resources: ["*"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: ["route.openshift.io"]
    resources: ["routes", "routes/custom-host"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]


---

kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ibm-transformation-advisor
  namespace: NAMESPACE_PLACEHOLDER
subjects:
  - kind: ServiceAccount
    name: ta-operator-sa
roleRef:
  kind: Role
  name: ibm-transformation-advisor
  apiGroup: rbac.authorization.k8s.io

---

apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: ibm-transformation-advisor
rules:
  - apiGroups: ["oauth.openshift.io"]
    resources: ["oauthclients", "oauthclients/finalizers", "consoleyamlsamples"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: ["console.openshift.io"]
    resources: ["consoleyamlsamples"]
    verbs: ["get", "list", "watch", "create", "update", "delete", "patch"]
  - apiGroups: ["config.openshift.io"]
    resources: ["ingresses", "infrastructures", "dnses"]
    verbs: ["get"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterrolebindings", "clusterroles", "clusterroles/finalizers",
"roles", "rolebindings", "roles/finalizers", "rolebindings/finalizers"]
    verbs: ["*"]
  - apiGroups: [""]
    resources: ["endpoints", "events"]
```

```
    verbs: ["get", "list", "watch", "create", "delete", "patch"]
  - apiGroups: ["discovery.k8s.io/v1"]
    resources: ["endpointslices"]
    verbs: ["get", "list", "watch", "create", "delete", "patch"]
  - apiGroups: ["network.openshift.io"]
    resources: ["netnamespaces", "clusternetworks", "egressnetworkpolicies"]
    verbs: ["get", "list", "watch", "create", "delete", "patch"]
---

kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: ibm-transformation-advisor-NAMESPACE_PLACEHOLDER
subjects:
  - kind: ServiceAccount
    name: ta-operator-sa
    namespace: NAMESPACE_PLACEHOLDER
roleRef:
  kind: ClusterRole
  name: ibm-transformation-advisor
  apiGroup: rbac.authorization.k8s.io
```

6. Install Operator

NOTE: For a disconnected install, install the operator using the ibm-pak-plugin for a disconnected/air-gap installation.

Install the operator using the Red Hat OpenShift UI

- Navigate the to *Operators...OperatorHub* in OpenShift navigation.
- Select the *IBM Operator Catalog* in the *Source* filter.
- Select the *Transformation Advisor* tile, filtering as necessary. Read the installation instructions and action as necessary.
- Click *Install*
- Choose the desired options and click *Install*

After a few minutes you should see confirmation that the operator has installed successfully. The Transformation Advisor operator is now available in the chosen namespace or all namespaces, depending on the install mode selected.

7. Create Entitlement Key

Complete the following steps to create a docker-registry secret to enable your deployment to pull operand images from the IBM® Entitled Registry.

This secret must be specified when installing the Transformation Advisor instance in step 9. See Configuration for more details.

Option 1: Create the entitlement key secret with the OpenShift console

1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the OpenShift console, ensure that you are in the Project (for example, my-project) that you installed the operator into.
3. Click Workloads > Secrets > Create, then select Image pull secret.

4. In the Secret name field, enter ibm-entitlement-key.
5. In the Registry server address field, enter cp.icr.io.
6. In the Username field, enter cp.
7. In the Password field, enter the credential that you received from entitlement.
8. (Optional) In the Email field, enter a contact email address.

Option 2: Create the entitlement key secret with the CLI

Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library Opens in a new tab with the IBMid and password details that are associated with the entitled software. In the Entitlement keys section, select Copy key to copy the entitlement key to the clipboard. From the Red Hat OpenShift CLI, run the following command to create an image pull secret called ibm-entitlement-key.

```
oc create secret docker-registry ibm-entitlement-key \
--docker-username=cp\
--docker-password=<entitlement-key> \
--docker-server=cp.icr.io \
--namespace=<namespace>
```

Where:

```
- `<entitlement-key>` is the entitlement key that you copied in step 2.
- `<namespace>` is the namespace that you want to install the Transformation
Advisor instance in.
```

As an alternative to creating the **`ibm-entitlement-key`** secret, you may update the cluster's global pull secret using your entitlement key credentials. See https://docs.openshift.com/container-platform/4.9/openshift_images/managing_images/using-image-pull-secrets.html#images-update-global-pull-secret_using-image-pull-secrets for how to update the global pull secret. Due to a limitation in the product, if you use the global pull secret, you still need to add the **`imagePullSecret`** properties to the Transformation Advisor instance configuration (See Configuration for more details). In this case, the value you use for the properties does not have to exist as a secret.

8. Perform Cluster Administration Tasks for Instance Installation

Before installing the Transformation Advisor instance, as **cluster administrator**, you will need to create a project for the Transformation Advisor installation.

Create a project to install the instance

If you have already chosen to install the Transformation Advisor Operator to a specific namespace, then you will have already created the project, otherwise create it now as follows:

Creating the project can be done from the OpenShift UI:

- Click the hamburger icon in the top left of the Red Hat OpenShift Container Platform dashboard.
- Expand **Home** menu and select **Projects**.
- Click **Create Project** button.
- Name the project and click **Create**.

Alternatively, you can use the **`oc new-project NAMESPACE_PLACEHOLDER`** command from the command line.

You will need the name of the project during the installation.

You can now install the Transformation Advisor instance as a project admin user.

9. Create Transformation Advisor instance

IMPORTANT: Please read the Cluster Admin Tasks section carefully before proceeding to install.

Please substitute NAMESPACE_PLACEHOLDER for the actual name of the project you are working with.

Find the Transformation Advisor operator in the OpenShift UI in your created project to configure and create an instance of Transformation Advisor.

- Click on Operators...Installed Operators
- From Projects drop-down select NAMESPACE_PLACEHOLDER
- Click on the Transformation Advisor operator
- Click on the **Create instance** link in the **Details** tab, or click into the **Transformation Advisor** tab and click **Create TransAdv**.

Configuring the instance

- **You will need to read and accept the license terms on the create instance page before you will be allowed to proceed with the install of the instance.**

- The default configuration gives you most of what you need to install the product. Click the **YAML View** radio button to access the custom resource YAML. Update values in that YAML as required.

- **You will need to configure the persistence before proceeding with the install**. This can be done from the UI form view, or the YAML view. Please see the Configure Storage document for full details, including examples, of the persistence configuration.

- You may also need to add image pull secrets to access Transformation Advisor images in the entitled registry. See Create Entitlement Key and Planning section for more details on accessing images in the entitled registry.

See Configuration page for a full list of the configurations available in the custom resource YAML file.

- When you are happy with your configuration, click **Create**.

Access the Transformation Advisor UI

By default, the Transformation Advisor operator exposes the UI as a Route on OpenShift. Go to **Networking - Routes** and click the location of the ui-route.

Validating a successful install

To validate the installation, on the OpenShift UI navigate to the pods for the projects where Transformation Advisor is installed. You should see the following fours pods in a READY and Running state:

```
ta-couchdb-0
ta-neo4j-0
ta-server-<ID>
ta-ui-<ID>
```

Additionally, you should see the instance pre-install job in a "Completed" state:

```
ta-operator-instance-preinstall-<ID>
```

If you have performed a single namespace install, then you will also see the operator pod in a READY and Running state:

```
ibm-transformation-advisor-manager-<ID>
```

Navigate to the Transformation Advisor UI as described here. Click on the kebab icon on the top right of the Transformation Advisor UI and then click on "What's New". Confirm the Transformation Advisor version in the What's New dialog.

Uninstalling Transformation Advisor

If you have installed Transformation Advisor using the Red Hat OpenShift UI, perform the following steps to completely remove the operator and instance. It is important not to attempt to delete the project before deleting the resources in the project.

1. Uninstall the instance:

    1. Click on Operators...Installed Operators in the left navigation.
    2. In the main panel, select the project dropdown and choose the project where Transformation Advisor instance is installed
    3. Click on the operator, and then on the **Transformation Advisor** tab
    4. Click on the kebab button for the listed instance and select **Delete Transadv**
    5. In less than 2 minutes, the instance should disappear (the associated pods may take longer to terminate). If the instance is still visible after two minutes, first try refreshing the browser. If the instance is still shown, please perform the following oc command (replace the NAMESPACE_PLACEHOLDER with the name of the project the instance is installed into):

```
oc patch transadvs.ta.ibm.com/ta -p '{"metadata":{"finalizers":[]}}' --
type=merge -n NAMESPACE_PLACEHOLDER
```

2. Uninstall the operator:

    1. Click on Operators...Installed Operators in the left navigation.
    2. In the main panel, select the project dropdown and choose the project where Transformation Advisor instance is installed
    3. Click on the kebab button for the listed operator and select **Uninstall operator**
    4. In less than 2 minutes, the operator should disappear (the associated pods may take longer to terminate). If the operator is still visible after two minutes, first try refreshing the browser. If the operator is still shown, please perform the following oc command:

```
oc patch crd/transadvs.ta.ibm.com -p '{"metadata":{"finalizers":[]}}' --
type=merge
```

3. For a single namespace operator install, remove the manually created resources.

    1. Use the OpenShift UI (Click on Home...Search in the left navigation) to find and delete the following resources:
        1. ServiceAccount called ta-operator-sa
        2. OperatorGroup called ta-ownnamespace-operator-group
        3. Role called ibm-transformation-advisor
        4. RoleBinding called ibm-transformation-advisor
        5. ClusterRole called ibm-transformation-advisor
        6. ClusterRoleBinding called ibm-transformation-advisor
    2. Alternatively, you can delete these resources from the command line using the **oc delete...** command.

4. Delete the project. It is recommended that you delete the project before attempting another install of Transformation Advisor
    1. Use the OpenShift UI (Click on Home...Projects) to delete the created Project

.

Upgrading

NOTE: Back up your data before performing an upgrade. See Backup / Export for more details.

Upgrading to Transformation Advisor 3.10.0 from 3.9.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.10.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.9 channel in the **Update channel** area. Select the v3.10 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.9.0 from 3.8.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.9.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.8 channel in the **Update channel** area. Select the v3.9 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.8.0 from 3.7.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.8.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.7 channel in the **Update channel** area. Select the v3.8 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.7.0 from 3.6.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.7.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.6 channel in the **Update channel** area. Select the v3.7 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.6.0 from 3.5.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.6.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.5 channel in the **Update channel** area. Select the v3.6 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.5.0 from 3.4.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.5.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.4 channel in the **Update channel** area. Select the v3.5 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.4.0 from 3.3.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.4.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.3 channel in the **Update channel** area. Select the v3.4 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.3.0 from 3.2.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.3.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.2 channel in the **Update channel** area. Select the v3.3 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.2.0 from 3.1.x

Back up your data before upgrading. See Backup / Export for more details.

To upgrade to Transformation Advisor 3.2.0, you will need to switch the subscription channel. In the OpenShift UI, navigate to the Transformation Advisor Operator. Click on the Subscription tab and click to edit the v3.1 channel in the **Update channel** area. Select the v3.2 channel from the list of options. Allow Transformation Advisor several minutes to update.

Upgrading to Transformation Advisor 3.1 from 3.0

Back up your data before upgrading. See Backup / Export for more details.

How to upgrade:

To upgrade to 3.1.0, you will need to switch the subscription channel. This happens via Subscription tab in the IBM Transformation Advisor operator page. Click on v3.0 in the **Update channel** area and select v3.1. The product will automatically upgrade after saving the change of the channel. If you have edited the

Transformation Advisor subscription to require manual approval for updates, you will have to confirm the upgrade.

Known issue:

There is a known issue when upgrading from 3.0.0 to 3.1.0 when Transformation is installed in a OpenShift cluster that does not support **`EgressNetworkPolicies`**. OpenShift clusters on IBM Cloud currently (correct at time of writing, April 2022) do not support **`EgressNetworkPolicies`**. In this case, the upgrade will not complete successfully. Follow this procedure to work around the issue:

1. Perform the upgrade by changing the subscription channel as normal.
2. After the operator has upgraded, the Transformation Advisor instance upgrade will stall. If you examine the logs of the operator pod, you will see a message like this:

```
...
no matches for kind \"EgressNetworkPolicy\"
...
```

3. Go to the instance (Click into the Transformation Advisor operator, then on the Transformation Advisor tab, and then on the instance in the table)
4. Click the YAML tab and update the **`networkPolicy`** section to disable egress policies. The **`networkPolicy`** should look like the following when you are finished your update:

```
  networkPolicy:
    enabled: true
    egress:
      enabled: false
```

5. Save the update and the upgrade should automatically continue and succeed.

Upgrading to Transformation Advisor 3.0 from 2.5.X and before

There is no upgrade path to Transformation Advisor 3.0.
We recommend that you upload your existing zip files again, or download the data collector and re-run it.
If you have existing data that only exists in Transformation Advisor then you may wish to migrate your data.
Further details on data migration can be found in our data migration section.
Further details on uninstalling 2.5.X and installing 3.0.X can be found in moving from 2.5.x to 3.0.x section

Upgrading to Transformation Advisor 2.5.0 from 2.4.4

This is NOT an automatic upgrade as the channel for Transformation Advisor 2.5.0 is different from 2.4.4. Switching the channel happens via Subscription tab in the IBM Transformation Advisor operator page. Click on v2.4 in the **Update channel** area and select v2.5. The product will automatically upgrade after saving the change of the channel. If you have edited the Transformation Advisor subscription to require manual approval for updates, you will have to confirm the upgrade.

Upgrading to Transformation Advisor 2.4.4 from 2.4.3

This is an automatic upgrade by default. Unless you have edited the Transformation Advisor subscription to require manual approval for updates, Transformation Advisor will automatically update to version 2.4.4 from 2.4.3. If you have edited the Transformation Advisor subscription to require manual approval for updates, check the OpenShift UI for install plans requiring approval.

Upgrading to Transformation Advisor 2.4.3 from 2.4.2

This is an automatic upgrade by default. Unless you have edited the Transformation Advisor subscription to require manual approval for updates, Transformation Advisor will automatically update to version 2.4.3 from 2.4.2. If you have edited the Transformation Advisor subscription to require manual approval for updates, check the OpenShift UI for install plans requiring approval.

KNOWN ISSUE UPGRADING TO 2.4.3

If you have used the Transformation Advisor version 2.4.2 CASE installer to upgrade from version 2.3.x to version 2.4.2, then the upgrade to 2.4.3 will not work. In the OpenShift UI, the Operator upgrade will appear to be stuck in a **`Pending`** state. If you encounter this problem, *and only if you have used the 2.4.2 CASE installer to upgrade from 2.3.4*, you need to execute the following procedure to fix Transformation Advisor:

1. Delete the Transformation Advisor subscription

   ```
   oc delete subscription ibm-transformation-advisor -n <namespace>
   ```

2. Delete the Transformation Advisor CSV

   ```
   oc get csv -n <namespace>
   ```

   ```
   oc delete csv <csv-name1> <csv-name2> -n <namespace>
   ```

3. Patch the Transformation Advisor instance configuration with license type. Visit this location to find the correct value for your installation.

   ```
   oc patch transadvs.ta.ibm.com/ta -p '{"spec":{"license":{"aLicenseType": "IBM
   Cloud Transformation Advisor 2.4.2 (Evaluation) - L-NHON-BYPFMS"}}}' --
   type=merge -n <namespace>
   ```

4. Recreate the Transformation Advisor subscription.

   a. Save the following yaml to a file. Update the namespace as appropriate.

   ```
   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-transformation-advisor
     namespace: <namespace>
   spec:
     channel: v2.4
     name: ibm-transformation-advisor
     installPlanApproval: Automatic
     source: ibm-transadv-catalog
     sourceNamespace: openshift-marketplace
    startingCSV: ta-operator.v2.4.3
   ```
   ```

   b. Apply the subscription.

   ```
   ```
   oc apply -f <subscriptionfile.yaml>
   ```
   ```

In the **Installed Operators** Openshift UI, you should now see the 2.4.3 Transformation Advisor operator installs successfully, and after a number of minutes, the 2.4.3 instance will be available for use.

Upgrading to Transformation Advisor 2.4.2 from 2.4.1

This is an automatic upgrade by default. Unless you have edited the Transformation Advisor subscription to require manual approval for updates, Transformation Advisor will automatically update to version 2.4.2 from 2.4.1. If you have edited the Transformation Advisor subscription to require manual approval for updates, check the OpenShift UI for install plans requiring approval.

Upgrading to Transformation Advisor 2.4.1 from 2.4.0

This is an automatic upgrade by default. Unless you have edited the Transformation Advisor subscription to require manual approval for updates, Transformation Advisor will have automatically updated to 2.4.1 form 2.4.0. If you have edited the Transformation Advisor subscription, to require manual approval for updates, check the OpenShift UI for install plans requiring approval.

Upgrading to Transformation Advisor 2.4.X from 2.3.X

Transformation Advisor 2.4.X is delivered in the IBM Operator Catalog. See Planning for more details. Transformation Advisor 2.3.X was delivered in the Red Hat catalog **and is now deleted from it**. Consequently, there is no direct upgrade path to Transformation Advisor 2.4.X. If want to upgrade from Transformation Advisor 2.3.X to 2.4.X, you have the following options:

- Option 1: If you have no customizations to the Transformation Advisor instance, and no requirement to keep data, then you can simply uninstall 2.3.X and install 2.4.X as described in the installation instructions.

  You can uninstall 2.3.X in the OpenShift UI as follows:

  1. Find the Transformation Advisor operator in the **Installed Operators**.
  2. Click through on the Transformation Advisor operator to find the instance. You will see the instance when you click the **Transformation Advisor** tab.
  3. Click on the kebab icon for the Transformation Advisor instance and select **Delete TransAdv**.
  4. Back on the Installed Operators page, click on the kebab icon for the Transformation Advisor operator and select **Uninstall Operator**.

- Option 2: Download the CASE installer for Transformation Advisor and install 2.4.X using the `--upgrade` option. This will preserve any customizations to the 2.3.X instance and your 2.3.X data. See the *Installing using the CASE installer* section in the installation documentation for more details.

  In case you have modified authentication.ocp.secretName value on install, you need to patch your secret before running the upgrade using following script:

```
#!/bin/bash

random_string(){
  rand=$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w ${1:-32} | head -n
1)
  echo -n $rand | openssl sha1 | sed 's/^.* //'
}

secret_name=<YOUR_SECRET_NAME>

NONADMINUSER=$( random_string | base64 )
NONADMINSECRET=$( random_string | base64 )
oc patch secret $secret_name -p '{"data":{"db_nonadmin_user":
"'"$NONADMINUSER"'"}}' --type=merge
```

```
oc patch secret $secret_name -p '{"data":{"db_nonadmin_secret":
"'"$NONADMINSECRET"'"}}' --type=merge
```

- Option 3: Follow the manual instructions described here to keep your 2.3.X data.

  1. Backup your data from a previously installed version. See Backup and Restore document for your options for doing that.

  2. You can uninstall Transformation Advisor 2.3.x, while preserving the **`PersistentVolumeClaim`** that it uses. That **`PersistentVolumeClaim`** can then be used with you new Transformation Advisor 2.4.X install. Preserving the **`PersistentVolumeClaim`** is the easiest way to keep your data when upgrading to 2.4.x.

     TO KEEP YOUR 2.3.X DATA BY PRESERVING THE `PersistentVolumeClaim` follow the *Preserving the PersistentVolumeClaim* procedure outlined in Backing up and Exporting Data. When following this procedure, you must install Transformation Advisor 2.4.X into the same namespace as the 2.3.x version.

     If you have other customizations to your Transformation Advisor instance (i.e. if you have edited other values in the custom resource YAML when installing 2.3.X), you may also wish to keep or re-apply them. You can view your current Transformation Advisor instance configuration with the following command:

     ```
     kubectl get transadvs.charts.ta.cloud.ibm.com ta -n <NAMESPACE> -o yaml
     ```

  3. In case you have modified the **`authentication.ocp.secretName`** value on install, run the script provided in Option 2.

  4. Install 2.4.X as per Installing document. If you have preserved the **`PersistentVolumeClaim`**, then you can use that as an **`existingClaim`** for your 2.4.X install.

.

Upgrading From Evaluation License

Overview

IBM Cloud Transformation Advisor can be installed on Red Hat OpenShift Container Platform and used for free for 90 days. After 90 days you will need entitlement to continue using the product. The following instruction should be followed when the 90 Day Free Evaluation period is over and you have purchased entitlements to a product containing Transformation Advisor.

Step 1: Get a key to the entitled registry.

An entitlement key for the software is associated with your MyIBM account. Get the entitlement key that is assigned to your ID.

- Log in to MyIBM Container Software Library with the IBMid and password that are associated with the entitled software.
- In the Entitlement keys section, select Copy key to copy the entitlement key to the clipboard.

Step 2: Create an image pull secret for the entitled registry

Here is a sample command to create the image pull secret. Substitute in your own values for the entitled registry user and key.

```
oc -n <NAME_SPACE> create secret docker-registry <SECRET_NAME> --docker-
server=cp.icr.io --docker-username=<ENTITLED_REGISTRY_USER> --docker-password=
<ENTITLED_REGISTRY_KEY>
```

Step 3: Update the Transformation Advisor instance.

Navigate to the Transformation Advisor instance in OpenShift Container Platform:

- Click on **Installed Operators** and set the project to the project where Transformation Advisor is installed
- Click through on the Transformation Advisor operator and then on the **Transformation Advisor** tab
- Click on the listed instance, and then click on the **YAML** tab
- Add the image pull secret for each Transformation advisor image as follows:

```
...
couchdb:
  imagePullSecret: <SECRET_NAME>
  ...
neo4j:
  imagePullSecret: <SECRET_NAME>
  ...
transadv:
  imagePullSecret: <SECRET_NAME>
  ...
transadvui:
  imagePullSecret: <SECRET_NAME>
  ...
...
```

- Click the **Save** button
- Allow several minutes for the Transformation Advisor containers to restart. After restarting, the containers will pull the images from the entitled registry. You can confirm by looking at the events for the containers to see an event which shows the entitled registry is used:

```
Successfully pulled image "cp.icr.io/cp/icpa/transformation-advisor-ui@sha256:
<actual image digest here>" in 1.007108858s
```

Using a global pull secret

You may also configure the entitled registry credentials in the global pull secret on the cluster. In this case, you still need to add the **imagePullSecret** properties to the Transformation Advisor instance as described in step 3. However, instead of specifying a real secret, you can specify an arbitrary string as the secret name. For example:

```
...
couchdb:
  imagePullSecret: dummytapullsecret
  ...
neo4j:
  imagePullSecret: dummytapullsecret
  ...
transadv:
  imagePullSecret: dummytapullsecret
  ...
transadvui:
  imagePullSecret: dummytapullsecret
  ...
...
```

.

Installing IBM Cloud Transformation Advisor in a Disconnected/Air-gapped Environment

If your cluster is not connected to the internet, you can install IBM Cloud Transformation Advisor in your cluster by using the ibm-pak-plugin and the CASE package.

NOTE: As of Transformation Advisor 3.3.0, it is recommended to use the ibm-pak-plugin tool as described here to perform the disconnected installation. The ibm-pak-plugin replaces the use of the cloudctl CLI tool that was used to perform the install in previous versions.

Overview

IMPORTANT: Please read the Cluster Administration Tasks section in the installation instructions before proceeding with the installation.

It is common in production to have a cluster that does not have internet access. This is often referred to as a disconnected, air-gapped or offline environment. This document describes how to install Transformation Advisor in such an environment using the ibm-pak-plugin and the CASE package.

About the CASE based installation

The Container Application Software for Enterprises (CASE) specification defines metadata and structure for packaging a containerized application. For more details please see CASE Specification.

The CASE installer provides the commands needed to install/uninstall Transformation Advisor using the CLI. The CASE install action will install both the operator and instance with a single command.

Installing using the CASE installer supports the same two install modes as installing using the OpenShift UI:

- All namespaces on the cluster (default)

  Operator will be available in all Namespaces.

- A specific namespace on the cluster (specify the **`--namespaceScoped true`** in the **`--args`**)

Operator will be available in a single Namespace only.

For both installation modes, you are required to perform some steps as a cluster administrator as described in the Cluster Administration Tasks section in the installation instructions.

The Transformation Advisor CASE contains three inventory items:

- v2InstallProduct - Provides the CASE actions to install/uninstall the Transformation Advisor product. It will first install the operator - optionally installing catalogs as required - and then install the instance.
- v2TransAdvOperator - Provides the CASE actions to install/uninstall the Transformation Advisor operator.
- v2TransAdv - Provides the CASE actions to install/uninstall the Transformation Advisor instance.

Each inventory has a README.md with details on the install actions and options available in that inventory.

About the ibm-pak-plugin

The ibm-pak-plugin is a plugin for the **`oc`** CLI tool that streamlines the deployment of IBM CloudPaks in a disconnected environment which was done earlier using [cloudctl] (https://github.com/IBM/cloud-pak-cli).

Please read carefully the ibm-pak-plugin documentation.

Installation Procedure

Catalog-based image mirroring

Starting with **`ibm-pak`** v1.8.0, the plug-in lays the foundation for eventual support for catalog-based mirroring. Information about catalog-based mirroring is described in the doc. At this time, catalog-based mirroring and oc-mirror tool usage is a Tech Preview feature, which may not be supported by all products. Transformation Advisor is enabled for catalog based mirroring starting from version 3.8.0. **`oc mirror`** tool usage is enabled in Transformation Advisor as a Tech Preview. The recommended install path continues to leverage **`oc image mirror`** as described in the steps here.

A consequence of enabling catalog based mirroring in Transformation Advisor is that the **`images-mapping.txt`** file generated by **`oc ibm-pak generate mirror-manifests ...`** command now contains all the versions necessary to install Transformation or to upgrade from any allowed upgrade path.

KNOWN ISSUE: In Transformation Advisor **3.8.0** and **3.8.1** there is a known issue whereby more images than are necessary are populated in the **`~/.ibm-pak/data/mirror/ibm-transadv/VERSION/images-mapping.txt`** after running **`oc ibm-pak generate mirror-manifests ...`**. Additionally, some of the image references are incorrect and cause the subsequent **`oc image mirror ...`** command to fail. You can manually edit the **`~/.ibm-pak/data/mirror/ibm-transadv/VERSION/images-mapping.txt`** file before running **`oc image mirror ...`** to remove unwanted images. See following note on removing all but current version images.

If installing Transformation Advisor for the first time, or if upgrading from Version 3.0.1+, you can use the following command to reduce the number of images to mirror (Note: you must have already run the **`oc ibm-pak generate mirror-manifests ...`** command):

Replace MAJOR, MINOR and PATCH, with the MAJOR, MINOR and PATCH numbers that correspond to the version of Transformation Advisor that you are installing.

```
cd ~/.ibm-pak/data/mirror/ibm-transadv/MAJOR.MINOR.PATCH/
mv images-mapping.txt images-mapping.txt.orig; grep ":MAJOR\.MINOR\.PATCH" images-mapping.txt.orig > images-mapping.txt; grep ":vMAJOR\.MINOR" images-mapping.txt.orig >> images-mapping.txt
```

Steps

1. Download the `ibm-pak` plugin, and `oc` tools. Downloading the tools is described in the ibm-pak documentation.
2. Configure the `ibm-pak` plugin to point to the CASE repository. You can choose to point to the IBM GitHub repo (https://github.com/IBM/cloud-pak/raw/master/repo/case/) **OR** (as of `ibm-pak` version 1.2.0) to the IBM Cloud Container Registry and download the CASE as OCI artifacts.

```
   oc ibm-pak config repo 'IBM Cloud-Pak Github Production Repo' --url
https://github.com/IBM/cloud-pak/raw/master/repo/case/ --enable
```

OR

```
oc ibm-pak config repo 'IBM Cloud-Pak OCI registry' -r oci:cp.icr.io/cpopen --
enable
```

3. Choose one of the supported paths for the disconnected installation. The IBM Cloud Pak documentation describes the two supported paths: `Bastion host` and `Filesystem`.
4. Download the Transformation Advisor CASE - substituting the desired version:

```
export CASE_NAME=ibm-transadv
export CASE_VERSION=X.Y.Z
oc ibm-pak get $CASE_NAME --version $CASE_VERSION
```

5. Verify you have downloaded the correct version:

```
oc ibm-pak list --downloaded
```

6. Generate the mirror manifests - substituting the actual target registry:

    1. For `Bastion host` path:

    ```
    export TARGET_REGISTRY=mytargetregistry.com
    oc ibm-pak generate mirror-manifests $CASE_NAME $TARGET_REGISTRY --
    version $CASE_VERSION
    ```

    2. For `Filesystem` path:

    ```
    export TARGET_REGISTRY=mytargetregistry.com
    oc ibm-pak generate mirror-manifests \
     $CASE_NAME \
     file://local \
     --version $CASE_VERSION \
     --final-registry $TARGET_REGISTRY
    ```

7. Authenticate to all registries. Follow the instructions here to authenticate to all registries using either `podman` or `docker`. Set the `REGISTRY_AUTH_FILE` environment variable appropriately.

8. (Bastion) Mirror images - `Bastion host` path:

```
oc image mirror \
 -f ~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION/images-mapping.txt \
 --filter-by-os '.*'  \
 -a $REGISTRY_AUTH_FILE \
 --insecure  \
 --skip-multiple-scopes \
 --max-per-registry=1
```

8 (Filesystem). Mirror images - `Filesystem` path:

    1. Download images to file system:

```
oc image mirror \
 -f ~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION/images-mapping-to-filesystem.txt \
 --filter-by-os '.*' \
 -a $REGISTRY_AUTH_FILE \
 --insecure \
 --skip-multiple-scopes \
 --max-per-registry=1
```

2. Move the generated items to a machine inside the firewall which has access to the TARGET_REGISTRY. The items to move are:
   1. v2 directory
   2. The auth file referred by **`$REGISTSRY_AUTH_FILE`**
   3. **`~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION/images-mapping-from-filesystem.txt`**
3. Move images to final destination:

```
oc image mirror \
 -f ~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION/images-mapping-from-filesystem.txt \
 -a $REGISTRY_AUTH_FILE \
 --from-dir=${v2_dir} \
 --filter-by-os '.*' \
 --insecure \
 --skip-multiple-scopes \
 --max-per-registry=1
```

`${v2_dir}` refers to the parent directory on the file system where the v2 directory was copied to.

9. [Optional] Configure cluster for insecure registry

   - If your TARGET_REGISTRY is insecure, you must add it to the cluster insecureRegistries list.

```
oc patch image.config.openshift.io/cluster --type=merge \
-p '{"spec":{"registrySources":{"insecureRegistries":
["'${TARGET_REGISTRY}'"]}}}'
```

10. Update the global image pull secret for your Red Hat OpenShift cluster. Add in the secret to allow your cluster to pull from your mirror registry - if you have not already set that up. Follow the steps in Updating the global cluster pull secret.

11. Create ImageContentSourcePolicy

```
oc apply -f  ~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION/image-content-source-policy.yaml
```

Wait until all nodes are ready. You can monitor the nodes with the following command:

```
oc get nodes -w
```

Alternatively, monitor the MachineConfigPool - and ensure the the **`UPDATED`** column read **`True`** for the relevant worker pool:

```
oc get MachineConfigPool -w
```

12. Install Transformation Advisor

   - The following command will perform a full installation of Transformation Advisor. It will first install the private catalog, then the Transformation Advisor operator, and finally the instance. Substitute your desired namespace, a valid license type and set other options as required.

```
export NAMESPACE=my_ta_namespace
oc ibm-pak launch \
 $CASE_NAME \
 --version $CASE_VERSION \
 --action install \
 --inventory v2InstallProduct \
 --namespace $NAMESPACE \
 --args "--acceptLicense true --licenseType X-XXXX-XXXXXX --installTaCatalog
--persistence true --storageClass ibmc-block-gold"
```

See install options for more details on the options accepted by Transformation Advisor.

Transformation Advisor installation options

The following options can be provided in the **`--args`** parameter to customize the installation.

- View the full set of options: **`--taHelp`**
- View the different license options available: **`--viewLicense`**
- Install the Transformation Advisor private catalog if it doesn't already exist. Required for disconnected install: **`--installTaCatalog`**
- Turn persistence on and set the storage class: **`--persistence true --storageClass ibmc-block-gold`**

For a full list of options see Configuring Transformation Advisor.

Accessing the UI

At the end of a successful install, you will be presented with a link in the console that will take you to the Transformation Advisor UI. Alternatively, you can navigate to the Transformatino Advisor route in the OpenShift UI by going to **Networking** - **Routes** and click the location of the ui-route. You will need to have a valid login to the cluster to access Transformation Advisor UI.

Validating a successful installation

To validate the installation, using either the OpenShift UI or the **`oc`** CLI tool to check the Transformation Advisor pods. You should see the following fours pods in a READY and Running state:

```
ta-couchdb-0
ta-neo4j-0
ta-server-<ID>
ta-ui-<ID>
```

Additionally, you should see the instance pre-install job in a "Completed" state:

```
ta-operator-instance-preinstall-<ID>
```

If you have performed a single namespace install, then you will also see the operator pod in a READY and Running state:

```
ibm-transformation-advisor-manager-<ID>
```

Navigate to the Transformation Advisor UI as described here. Click on the kebab icon on the top right of the Transformation Advisor UI and then click on "What's New". Confirm the Transformation Advisor version in the What's New dialog.

Uninstalling with ibm-pak-plugin

IMPORTANT: If the CASE installation fails for any reason (if for example you set an install argument incorrectly like the user or password) and you want to retry the installation, first uninstall using the CASE uninstall command that is shown below. Do not manually delete the target namespace before you have run the uninstall command.

Run the following command to uninstall:

```
oc ibm-pak launch \
 $CASE_NAME \
 --version $CASE_VERSION \
 --action uninstall \
 --inventory v2InstallProduct \
 --namespace $NAMESPACE
```

Upgrading in an disconnected environment

Note: It is not possible to upgrade from Transformation Advisor 2.5.x to 3.0.x. You must uninstall 2.5.x before then installing 3.0.x. Please read Moving from 2.5.x to 3.0.x and data migration for more details.

To perform an upgrade in an disconnected/air-gapped environment, perform the steps described above to mirror the Transformation Advisor images for the latest release to your mirror registry. After that is completed, update the Transformation Advisor catalog to the latest version with the following command:

```
oc ibm-pak launch \
 $CASE_NAME \
 --version $CASE_VERSION \
 --action install-catalog \
 --inventory v2TransAdvOperator \
 --namespace $NAMESPACE
```

After the catalog is updated, the Transformation Advisor operator will become ready for upgrade. This might take some time and depends on the Registry Poll Interval for the catalog. By default that is set to 45 minutes.

If automatic upgrade is available for the release, Transformation Advisor will automatically upgrade. If you have disabled automatic upgrades, you must accept the upgrade in the OpenShift UI. Otherwise, you might need to subscribe to a new channel to receive the upgrade. For release specific instructions for upgrading please see upgrading.

TODO: If you are installing the evaluation images it is necessary to update the image locations to point to the registry icr.io/appcafe instead of the entitled registry. This step is completed as follows: If you are installing the evaluation images it is necessary to rename the ImageContentSourcePolicy. This step is completed as follows: **`oc get imagecontentsourcepolicy ibm-transadv -o yaml > saved_icsp.yaml; oc delete imagecontentsourcepolicy ibm-transadv; sed "s/ibm-transadv/ibm-eval-transadv/g" saved_icsp.yaml | oc apply -f - ; rm -f saved_icsp.yaml`**```

.

Installing IBM Cloud Transformation Advisor on RHEL

SUPPORT TYPE: Transformation Advisor is fully supported for installation on Red Hat Enterprise Linux. Limited Support is available for installation on Windows and is provided on a best-efforts basis. Security features such as authentication, TLS and proxy configuration are not supported on Windows.

Prerequisites

Before you begin go to the Registration and download site and download the Transformation Advisor Local install script. If you are using a Windows operating system, you must also download the **`docker-compose.yml`** and **`.env`** files. For details on using the signature to verify the downloaded zip, please see here.

Required Resources

Table 1. Minimum default configuration

```
System                    | Memory (GB) | CPU (cores) | Disk space (GB)
------------------------- | ----------- | ----------- | ---------------
Transformation Advisor    | 8           | 4           | 20
```

Installing IBM Transformation Advisor on RHEL

You might also follow these instructions for an installation on MacOS, subject to the **SUPPORTED PLATFORMS** note on this page regarding support.

Before you start, ensure that you have docker or podman installed.

- Podman greater than 3.4
- Docker on RHEL 7
- Docker on RHEL 8/9:

  Docker is not shipped or supported by Red Hat for Red Hat Enterprise Linux (RHEL) 8/9. Docker Community Edition may be installed on RHEL 8/9 from the Docker CE package repository here: https://download.docker.com/linux/centos/docker-ce.repo

1. If you haven't already, go to the Registration and download site and download the Transformation Advisor Local install script.
2. Create a directory for the Transformation Advisor files, for example, **`ta_local`**. Copy the .zip file that you downloaded during the registration step into this directory and extract it: **`unzip transformationAdvisor.zip`**
3. To install Transformation Advisor locally, run the following command: **`./launchTransformationAdvisor.sh`**
4. Select 1 if you agree with the terms of the License.
5. Select 1 to install Transformation Advisor
6. After the installation is complete, you can access Transformation Advisor locally at the following URLs. The host name or IP address and port are provided by the installation program.
   - Linux: **`http://< host name >:3000`**
   - MacOS: **`http://< IP Address >:3000`**

Note: For Transformation Advisor to function correctly your system must be configured to allow ingress and egress on ports 3000 & 2220

Installing IBM Cloud Transformation Advisor on Windows v10 / v11

Installation on Windows is subject to the **SUPPORTED PLATFORMS** note on this page.

1. If you haven't already, go to the Registration and download site and download the windows10files.zip.
2. Install Docker Desktop for Windows. You can watch this short video for a detailed walk-through of the installation steps.
3. Extract the zip and if Docker Desktop uses volumes then rename the file **`docker-compose_volumes.yaml`** to **`docker-compose.yml`**, if Docker Desktop uses file share then rename the file **`docker-compose_fileShare.yaml`** to **`docker-compose.yml`**
4. Create a **`ta_local`** directory, for example **`C:\Users\ta_local\dockerCompose`**, and put the **`.env`** and **`docker-compose.yml`** files there.
5. Open a terminal session.
6. Change to the directory where the **`docker-compose.yml`** and **`.env`** files are located.
7. Make sure that Docker is running:

   ```
   docker ps
   ```

8. Pull the Transformation Advisor images:

   ```
   docker-compose pull
   ```

9. Start the containers and run them in the background:

   ```
   docker-compose up -d
   ```

10. Verify that four containers are created:

`docker ps`

11. Access the Transformation Advisor UI at the following URL:

    ```
    http://localhost:3000
    ```

Installing IBM Cloud Transformation Advisor on Windows v10 / v11 using podman

1. Install Windows Terminal
2. Install / enable WSL 2
3. Install Ubuntu (20.04)
4. Install Podman from the **Add the Podman PPA and install Podman** section
5. Install TA using the script transformation-advisor-local-<version>.zip

https://www.redhat.com/sysadmin/podman-windows-wsl2

Known issue

On a reboot, you might see an error message like
`error retrieving network namespace`

https://github.com/containers/podman/issues/12236

Workaround

```
cd /tmp
rm -rf
```

Run the launch script again and select 5 to start the Transformation Advisor.

.

Security Hardening (non OCP install)

After installing you can optionally complete this security hardening step.
By default the docker and podman containers allow egress.
You can run the script provided to stop all egress except on the specified ports.
You may also configure additional egress restrictions or allowances by adding your own rules to the provided script

Executing the security hardening script

NOTE: You must have root permissions to execute this script

1. Go to the install location

If you are using docker run this command **`cd scripts sudo ./hardenSecurityDocker.sh`**

If you are using podman run this command **`cd scripts sudo ./hardenSecurityPodman.sh`**

Egress will now only be allowed by the exceptions listed in the script

.

Configuring Authentication (non OCP install)

To configure authentication, you need to have an OAuth2 Server available, and complete these steps:

- Go to the install location and stop TA

```
./launchTransformationAdvisor.sh
Choose Stop Transformation Advisor from the menu option
```

- Now create the configuration

```
cd scripts
vi .security_config
Copy the environment variables (without comments i.e. any string after #) from the
`.security_config_sample` file. Replace the sample values with the actual values
according to the OAuth2 Server you are using.
```

- Start TA

```
./launchTransformationAdvisor.sh
Choose Start Transformation Advisor from the menu option
```

Sample configuration for `box.com` OAuth2 server

The following sample is the configuration for the box.com OAuth2 server.

```
TA_AUTH_ENABLE_TLS=false
TA_LOCAL_INTERNAL_SERVER_PORT=9080
TA_LOCAL_INTERNAL_UI_PORT=3000
TA_LOCAL_INTERNAL_DB_PORT=5984

TA_AUTH_UI_DISABLED=false
TA_AUTH_LIBERTY_DISABLED=false

TA_AUTH_OIDC_CLIENT_ID=lanjmi4l41orezoiz5ektitgnfvtoi4n
TA_AUTH_OIDC_CLIENT_SECRET=1N1CRuTdxlOM5c5XvH3PT3Ijv5Eeaf9r

TA_AUTH_IDENTITY_REQUEST_ENDPOINT=https://account.box.com
TA_AUTH_IDENTITY_REQUEST_ENDPOINT_PATH=/api/oauth2/authorize
TA_AUTH_IDENTITY_REQUEST_ENDPOINT_SCOPE=root_readonly
TA_AUTH_CALLBACK_STATE_PREFIX_PADDING=

TA_AUTH_TOKEN_REQUEST_ENDPOINT=https://api.box.com
TA_AUTH_TOKEN_REQUEST_ENDPOINT_PATH=/oauth2/token

TA_AUTH_TOKEN_VERIFICATION_ENDPOINT=https://api.box.com
TA_AUTH_TOKEN_VERIFICATION_ENDPOINT_PATH=/2.0/users/me

TA_AES_IV=
TA_AES_KEY=
```

Please note, some OAuth Apps do not allow non-https callback URLs i.e. the value of `TA_AUTH_OIDC_CALLBACK_URI` may not allow to use a non-https endpoint.

.

Configuring TLS (non OCP install)

- It is recommended that you configure to use your own certificate for TLS.
- If you do not provide a certificate then TA will generate a self-signed certificate when TLS is enabled.
  - There are a number of limitations when using a self-signed certificate.
  - We do not support LibreSSL.
- TA uses `.crt` format for public certification, and `.pem` format for private key.

Steps to enable TLS (non OCP install)

To configure TLS you need to update a number of environment variables in .security_config

- Create the configuration

```
cd scripts
vi .security_config
```

- Set the values in the file

```
TA_AUTH_ENABLE_TLS=true
TA_LOCAL_INTERNAL_SERVER_PORT=9443
TA_LOCAL_INTERNAL_UI_PORT=3443
TA_LOCAL_INTERNAL_DB_PORT=6984

TA_AES_IV=
TA_AES_KEY=
```

- Re-initialize TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Re-initialize Configuration from the menu option
```

Using your own certificate with TA_LOCAL

To use your own certificate you must have a **certificate** and a **private key**.
All encoding must be in the **pem** format.

You can use files directly or store the relevant values as environmental variables.

Certificates as files

Note: The certificate must be in a file called: **public.crt**.
Note: The private key must be in a file called: **private.pem**.
Note: The **private.pem** file must be unencrypted. If, when you open the file, it indicates that it is encrypted it must be decrypted before use

Complete the following steps:

- Go to the <TA_LOCAL_HOME> location and stop TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Stop Transformation Advisor from the menu option
```

- Configure TA_LOCAL to use your files

```
cd key
Copy the private.pem and public.crt files to this location
```

- Start TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Re-initialize Configuration from the menu option
```

Certificates as environmental variables

Note: Any files in the **key** folder (detailed above) will override these environmental variables.
To use environmental variables you must delete the **key** folder if it exists

Note: The **private.pem** file must be unencrypted. If, when you open the file, it indicates that it is encrypted it must be decrypted before use

Complete the following steps:

- Go to the <TA_LOCAL_HOME> location and stop TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Stop Transformation Advisor from the menu option
```

- Configure TA_LOCAL to use environmental variables

```
base64 -w 0 key/private.pem > key/private-base64
Copy the text in private-base64 and set it as the value for TA_PRIVATE_KEY in
.security_config
base64 -w 0 key/public.crt > key/public-base64
Copy the text in public-base64 and set it as the value for TA_PUBLIC_KEY in
.security_config
```

- Start TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Re-initialize Configuration from the menu option
```

Browser limitations when using self signed certificates

- Different browsers react differently when they encounter self-signed certificates
- Their behaviour is also dependant on the security settings for each user
- At the time of writing this is the current behaviour of different browsers

Chrome

- This will not allow you to connect to a server using a self-signed certificate
- Potential Solutions
  - Use your own certificate
  - Install the self-signed certificate as a trusted certificate in the browser

Firefox

- This will allow you to connect to a server using a self-signed certificate only after you accept the risk
- Note: You will not be able to upload data directly from the browser even after you accept the risk
- Potential Solutions
  - Use your own certificate
  - Install the self-signed certificate as a trusted certificate in the browser
  - Configure a proxy for TA, this will allow you to upload data directly from the browser

Safari

- This will allow you to connect to a server using a self-signed certificate only after you accept the risk
- Potential Solutions
  - Use your own certificate
  - Install the self-signed certificate as a trusted certificate in the browser

.

Configuring a proxy server (non OCP install)

TA can be configured to run behind a proxy server to support load balancing and enhanced security.
Note: Once configured to use a proxy only the proxy URL will be supported. Alternative URLs (machine ip etc) will no longer be supported.

TA configuration

You need to set the externally accessible URL to those of the proxy server.
You need to make these changes in **.env** and **.env_orig** files.

- Go to the install location and stop TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Stop Transformation Advisor from the menu option
```

- Now update the configuration

```
cd scripts
vi .env
```

- Update the following variables with the appropriate values based on your proxy server
  - [PROXY_URL] : The URL the proxy server is listening at. You must enter a value for this placeholder.
  - [PROXY_PORT] : The port the proxy server is listening at. You must enter a value for this placeholder.

```
TA_LOCAL_PUBLIC_ACCESSIBLE_API_SERVER_URL=<protocol>://[PROXY_URL]:[PROXY_PORT]
TA_PUBLIC_ACCESSIBLE_UI_URL=<protocol>://[PROXY_URL]:[PROXY_PORT]
```

```
vi .env_orig
```

- Update the following variables with the appropriate values based on your proxy server
  - [PROXY_URL] : The URL the proxy server is listening at. You must enter a value for this placeholder.
  - [PROXY_PORT] : The port the proxy server is listening at. You must enter a value for this placeholder.

```
TA_LOCAL_PUBLIC_ACCESSIBLE_API_SERVER_URL=<protocol>://[PROXY_URL]:[PROXY_PORT]
TA_PUBLIC_ACCESSIBLE_UI_URL=<protocol>://[PROXY_URL]:[PROXY_PORT]
```

- If you have enabled authentication then you also need to update the callback URI

```
vi .security_config
TA_LOCAL_TA_AUTH_OIDC_CALLBACK_URI=<protocol>://[PROXY_URL]:
[PROXY_PORT]/auth/callback
```

- Now start TA_LOCAL

```
./launchTransformationAdvisor.sh
Choose Start Transformation Advisor from the menu option
```

Proxy server configuration

The specific steps that you need to take will depend on the proxy server you are using.
For TA_LOCAL to work you need to ensure the proxy server supports the following:

- Proxy to the UI port (3000 or 3443) and the Advisor port (2220)
- Maximum upload size for files to be large enough for your data, the recommended default value is 250MB
- UI proxy must support web sockets
- If TLS is enabled you must support ssl and configure your proxy with the appropriate certificate

Sample configuration for nginx

A sample configuration is provided here to show the **minimum** configuration that is required for ngnix. Further configuration should be applied based on your specific environment and requirements.
In this example: - The proxy server is available at **proxy.example.com** - TA_LOCAL is available at **ta.example.com**

Sample Non TLS configuration

```
server {
   listen       3000;
   server_name  proxy.example.com;
   client_max_body_size 250M; #Max upload size

   location /lands_advisor/ {
     proxy_pass     http://ta.example.com:2220;
   }

   location / {
     proxy_pass     http://ta.example.com:3000;
     #Support websockets
     proxy_http_version 1.1;
     proxy_set_header Upgrade $http_upgrade;
     proxy_set_header Connection "upgrade";
   }
 }
```

Sample TLS configuration

This sample assumes the proxy has generated it's own self-signed certificate stored in snippets

```
server {
   listen       3443 ssl;

   include snippets/self-signed.conf;

   server_name  proxy.example.com;
   client_max_body_size 250M; #Max upload size

   location /lands_advisor/ {
     proxy_pass     https://ta.example.com:2220;
   }

   location / {
     proxy_pass     https://ta.example.com:3443;
     #Support websockets
     proxy_http_version 1.1;
     proxy_set_header Upgrade $http_upgrade;
     proxy_set_header Connection "upgrade";
   }
 }
```

The self-signed.conf file references the location of the certificate and the key.
This is a sample self-signed.conf file

```
ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;
```

.

Configuring Ports (non OCP install)

Default ports

The default ports for TA_LOCAL are as follows

- User Interface: 3000
- Advisor server: 2220

Changing Ports

You can change the default ports by following the steps below

1. Go to the scripts directory
   `cd <TA_LOCAL_HOME>/scripts`
2. Edit the .configuration file
   `vi .configuration`
3. Update the following variables with the port number you want

```
TA_EXTERNAL_PORT_SERVER=2220
TA_EXTERNAL_UI_PORT=3000
```

4. Run the launch script
   `<TA_LOCAL_HOME>/launchTransformationAdvisor`
5. Select Option 8 to reconfigure the system

When this completes TA_LOCAL will be running on the new ports you have selected

.

Finding your API user token (non OCP install)

If authentication is enabled, you need to provide the API user token to access the APIs. This will vary depending your local OAuth system.

For example on OCP, you need to do the following:

- Launch the OpenShift Cloud Platform Console UI.
- Click on the user dropdown and select **Copy Login Command**.
- Click **Display Token**.
- Copy the value of the `token` attribute.

Or on Box, you get the API token as follows:

- Go to the Box developer console and select the application for which to create a Developer Token.
- From the sidebar, select **Configuration**.
- In the **Developer Token** section, select **Generate Developer Token**.

Enter this value into the apiKey field in the OpenAPI UI.

.

Upgrade Instructions

1. Backup your existing data. Choose the option to export all your zip files via the bulkExport found here: https://www.ibm.com/docs/en/cta?topic=started-exporting-importing-data-via-http
2. Download the latest version of transformationAdvisor.zip from here: https://www.ibm.com/support/pages/node/6958773/
3. Uninstall Transformation Advisor
   - `./launchTransformationAdvisor.sh`
   - Choose Option 2 (Uninstall but leave the data)
4. Check the images have been removed
   - `docker images`
   - There should be no Transformation Advisor images present
5. Explode the latest transformation-advisor-local.zip you downloaded into a new location
6. cd into this exploded location
7. Restore the data to the new location
   - `cp -a <oldLocation>/data .`
   - `cp -a <oldLocation>/graph_data .`
   - `cp -a <oldLocation>/scripts/.neo4j_pass scripts/`
8. In the new location install Transformation Advisor
   - `./launchTransformationAdvisor.sh`
   - Accept the license agreement
   - Choose Option 1 (Install)
9. Confirm the upgrade
   - Navigate to the provided URL
   - You should see a list of your existing workspaces
   - Click the three vertical dots and select **What's New** to show the version of Transformation Advisor

Upgrading to Transformation Advisor 3.0 from 2.5.X

There is no upgrade path to Transformation Advisor 3.0.
We recommend that you upload your existing zip files again or download the data collector and re-run it.

Note: Do not uninstall your existing version of Transformation Advisor until you have exported any data necessary for your migration.

Complete the following steps to migrate your data:

1. Export the data you currently have in Transformation Advisor Use the bulk export API in TA 2.5 /advisor/v2/collectionArchives/bulkExport /advisor/v2/workspaces/{workspaceId}/collectionArchives/bulkExport

2. Uninstall Transformation Advisor by selecting **option 2 Uninstall Transformation Advisor (keep database data)**.

Install Transformation Advisor
3. Install Transformation Advisor 3.0 into a different location

Import your data
4. Import your exported data into Transformation Advisor 3.0 using the bulkImport API /advisor/v2/collectionArchives/bulkImport the bulk import API for TA 3.0

Upgrading Transformation Advisor on RHEL

A video showing the upgrade process is available here: https://www.youtube.com/watch?v=xMc83VUzH-8&t=110s

Please Note : As of Transformation Advisor release 2.5.0, the images for the trial version of the product have been migrated from Docker Hub to an IBM registry icr.io/appcafe. Therefore, if you already have a non-OCP (aka TA Local) installation of version 2.4.4 or earlier, the **Check for latest Transformation Advisor** option won't detect that the 2.5.0 release is available. Therefore, you must re-download the install scripts from the registration and download site and run the scripts as follows.

.

Upgrading (non-ocp only)

Upgrading from evaluation to productive usage

IBM Cloud Transformation Advisor can be installed and used for free for 90 days. After 90 days you will need entitlement to continue using TA_Local. The following instruction should be followed when the 90 Day Free Evaluation period is over and you have purchased entitlements to a product containing Transformation Advisor.

Step 1: Get a key to the entitled registry.

An entitlement key for the software is associated with your MyIBM account. Get the entitlement key that is assigned to your ID.

- Log in to MyIBM Container Software Library with the IBMid and password that are associated with the entitled software.
- In the Entitlement keys section, select Copy key to copy the entitlement key to the clipboard.

Step 2: Login to the Entitled Registry

To set the entitled registry information export the entitled registry, user and key.

```
export ENTITLED_REGISTRY=cp.icr.io
export ENTITLED_REGISTRY_USER=cp
export ENTITLED_REGISTRY_KEY=<entitlement_key>
```

Log in to the entitled registry with the following docker login command:

```
docker login "$ENTITLED_REGISTRY" -u "$ENTITLED_REGISTRY_USER" -p
"$ENTITLED_REGISTRY_KEY"
```

Step 3: Update the image tag in the .configuration script

- Go to the scripts directory
- Make a backup of the .configuration script
- Edit the .configuration script
- Modify the image tags
    - TA_LOCAL_CONFIG_COUCH_IMAGE=cp.icr.io/cp/icpa/transformation-advisor-db:<version>

    - TA_LOCAL_CONFIG_SERVER_IMAGE=cp.icr.io/cp/icpa/transformation-advisor-server:<version>
    - TA_LOCAL_CONFIG_UI_IMAGE=cp.icr.io/cp/icpa/transformation-advisor-ui:<version>
    - TA_LOCAL_CONFIG_NEO4J_IMAGE=cp.icr.io/cp/icpa/transformation-advisor-neo4j:<version>
- Save the file

Step 4: Run the installation script

- Go to TA_LOCAL home
- Run the following command

```
./launchTransformationAdvisor.sh
```

Upgrading to newer version

- From the existing TA local, copy the data directory to a temp location.
- Get the latest zip, and extract it
- Copy the data directory from the temp location to the location of the new ta local (ensure ownership and permissions do not get changed)
- Install the new TA version using the launchTransformationAdvisor.sh script.

.

Non OCP install airgap instructions

Non OCP airgap install instructions

On a system that has docker or podman installed and has internet connectivity, pull the Transfromation Advisor images. Then, save the images to a gzip'd tar ball.

```
docker / podman pull icr.io/appcafe/transformation-advisor-db:<version>
docker / podman pull icr.io/appcafe/transformation-advisor-ui:<version>
docker / podman pull icr.io/appcafe/transformation-advisor-server:<version>
docker / podman pull icr.io/appcafe/transformation-advisor-neo4j:<version>
```

If you using docker, run the command below

```
docker save icr.io/appcafe/transformation-advisor-db:<version>
icr.io/appcafe/transformation-advisor-ui:<version> icr.io/appcafe/transformation-
advisor-server:<version> icr.io/appcafe/transformation-advisor-neo4j:<version> |
gzip > ta-images.tar.gz
```

If you using podman, run the steps below

```
podman save -m icr.io/appcafe/transformation-advisor-db:<version>
icr.io/appcafe/transformation-advisor-ui:<version> icr.io/appcafe/transformation-
advisor-server:<version> icr.io/appcafe/transformation-advisor-neo4j:<version>
k8s.gcr.io/pause:3.5 | gzip > ta-images.tar.gz
```

On the Air Gap system, do the following steps:

```
Edit the file /usr/share/containers/containers.conf
Search for infra_image
uncomment this and add the following
infra_image = "k8s.gcr.io/pause:3.5"
Save the file
```

Transfer the file to the target server and run the launchTransformationadvisor script. Choose the option to work in an air gapped environment and you will be prompted to enter the location of the tarball containing the images.

.

Configuring IBM Cloud Transformation Advisor

When you install Transformation Advisor default values are applied for the configuration settings. Persistence is one configuration that you will need to set yourself depending on the persistence options available on your environment. Please refer to the Configure storage document for more details on that. Depending on your environment and preferences you may wish to customize your Transformation Advisor install further. The following is a list of all the configuration options available in Transformation Advisor, which can accessed in the custom resource YAML. See Installing on how to access the custom resource YAML. Many of these options (as indicated) are intended for advanced troubleshooting and should only be used by experts.

Parameter	Description	Default
networkPolicy.enabled	Enables networkPolicy in the cluster. Advanced troubleshooting.	true
networkPolicy.egress	Enable and configure the EgressNetworkpolicy. Auto disablewhen OpenShift SDN CNI is not used.	enabled, with default egress targets allowed.
route.enabled	Enables route to reach the service. Advanced troubleshooting.	true
route.hosname	Hostname for route.	Discovered and set by TA operator
tls.enabled	Enables TLS between containers. Advanced troubleshooting.	true
tls.caCert	CA certificate for TLS (see instructions on creating customer cert)	set by icpa-installer
authentication.disabled.liberty	Disable authentication for Liberty server. Advanced troubleshooting.	false
authentication.disabled.ui	Disable authentication for UI. Advanced troubleshooting.	false
authentication.ocp.authIssuerEndpoint	Authentication issuer endpoint.	Discovered and set by TA operator

Parameter	Description	Default
authentication.ocp.apiEndpoint	Authentication API endpoint.	Discovered and set by TA operator
authentication.ocp.secretName	Secret name for internal authentication.	transformation-advisor-secret
authentication.oidc.endpointPort	OIDC authentication endpoint port	Discovered and set by TA operator
authentication.thirdparty.identityRequestEndpoint	Third party identity request endpoint	Not set
authentication.thirdparty.identityRequestEndpointPath	Third party identity request endpoint path	Not set
authentication.thirdparty.identityRequestEndpointScope	Third party identity request endpoint scope	Not set
authentication.thirdparty.identityRequestEndpointStatePrefix	Third party identity request endpoint state prefix	Not set
authentication.thirdparty.tokenRequestEndpoint	Third party token request endpoint	Not set
authentication.thirdparty.tokenRequestEndpointPath	Third party token request endpoint path	Not set
authentication.thirdparty.tokenVerificationEndpoint	Third party verification endpoint	Not set
authentication.thirdparty.tokenVerificationEndpointPath	Third party verification endpoint path	Not set
couchdb.image	Couchdb image tag	Discovered and set by TA operator
couchdb.imagePullSecret	Image pull secret. Used to access entitled registry. Name must be ibm-entitlement-key. See Planning for more details.	None

Parameter	Description	Default
couchdb.security.cipherSuites	Cipher suites that should be supported (whitespace separated)	ECDHE-RSA-AES256-GCM-SHA384 ECDHE-RSA-AES128-GCM-SHA256 TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256
couchdb.security.tlsVersions	List of permitted SSL/TLS protocol versions (whitespace separated)	tlsv1.2
couchdb.resources.requests.memory	Requests memory (Resources could be modified depending on availability. Defaults are minimim.)	1Gi
couchdb.resources.requests.cpu	Requests cpu	500m
couchdb.resources.limits.memory	Limits memory	8Gi
couchdb.resources.limits.cpu	Limits cpu	16000m
couchdb.livenessProbe.initialDelaySeconds	Container liveness probe initial delay seconds (Probes could be modified depending on performance of the cluster. Defauls work in most cases though. Advanced troubleshooting.)	60
couchdb.livenessProbe.timeoutSeconds	Container liveness probe timeout seconds. Advanced troubleshooting.	3
couchdb.livenessProbe.periodSeconds	Container liveness probe period seconds. Advanced troubleshooting.	5
couchdb.livenessProbe.failureThreshold	Container liveness probe failure threshold. Advanced troubleshooting.	6
couchdb.readinessProbe.initialDelaySeconds	Container readiness probe initial delay seconds. Advanced troubleshooting.	5
couchdb.readinessProbe.timeoutSeconds	Container readiness probe timeout seconds. Advanced troubleshooting.	3

Parameter	Description	Default
couchdb.readinessProbe.periodSeconds	Container readiness probe period seconds. Advanced troubleshooting.	5
couchdb.readinessProbe.failureThreshold	Container readiness probe failure threshold. Advanced troubleshooting.	6
neo4j.image	NEO4J image reference	Discovered and set by TA operator
neo4j.imagePullSecret	Image pull secret. Used to access entitled registry. Name must be ibm-entitlement-key. See Planning for more details.	None
neo4j.resources.requests.memory	Requests memory (Resources could be modified depending on availability. Defaults are minimim.)	1Gi
neo4j.resources.requests.cpu	Requests cpu	500m
neo4j.resources.limits.memory	Limits memory	8Gi
neo4j.resources.limits.cpu	Limits cpu	16000m
neo4j.livenessProbe.initialDelaySeconds	Container liveness probe initial delay seconds (Probes could be modified depending on performance of the cluster. Defauls work in most cases though. Advanced troubleshooting.)	60
neo4j.livenessProbe.timeoutSeconds	Container liveness probe timeout seconds. Advanced troubleshooting.	3
neo4j.livenessProbe.periodSeconds	Container liveness probe period seconds. Advanced troubleshooting.	5
neo4j.livenessProbe.failureThreshold	Container liveness probe failure threshold. Advanced troubleshooting.	6

Parameter	Description	Default
neo4j.readinessProbe.initialDelaySeconds	Container readiness probe initial delay seconds. Advanced troubleshooting.	5
neo4j.readinessProbe.timeoutSeconds	Container readiness probe timeout seconds. Advanced troubleshooting.	3
neo4j.readinessProbe.periodSeconds	Container readiness probe period seconds. Advanced troubleshooting.	5
neo4j.readinessProbe.failureThreshold	Container readiness probe failure threshold. Advanced troubleshooting.	6
persistence.enabled	Persistence enabled (If disabled all the data will be lost if DB container restarts).	true
persistence.couchdb.accessMode	Couchdb access mode.	ReadWriteOnce
persistence.couchdb.size	Couchdb storage size.	8Gi
persistence.couchdb.useDynamicProvisioning	Use dynamic provisioning. Do not change.	true
persistence.couchdb.existingClaim	Existing pv claim (Usually existing PVC is used to point to existing data.)	""
persistence.couchdb.storageClassName	Couchdb storage class name (e.g. "rook-ceph-cephfs-internal")	""
persistence.couchdb.supplementalGroups	Couchdb supplemental groups (Usually used for NFS)	[]
persistence.neo4j.accessMode	Couchdb access mode.	ReadWriteOnce
persistence.neo4j.size	Couchdb storage size.	8Gi
persistence.neo4j.useDynamicProvisioning	Use dynamic provisioning. Do not change.	true
persistence.neo4j.existingClaim	Existing pv claim (Usually existing PVC is used to point to existing data.)	""

Parameter	Description	Default
persistence.neo4j.storageClassName	Couchdb storage class name (e.g. "rook-ceph-cephfs-internal")	""
persistence.neo4j.supplementalGroups	Couchdb supplemental groups (Usually used for NFS)	[]
transadv.image	Transadv Liberty server image tag. Advanced troubleshooting.	
transadv.imagePullSecret	Image pull secret. Used to access entitled registry. Name must be ibm-entitlement-key. See Planning for more details.	
transadv.security.cipherSuites	Cipher suites that should be supported (whitespace separated)	TLS_AES_256_GCM_SHA384 TLS_CHACHA20_POLY1305_SHA256 TLS_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
transadv.pubilcUrl	Transadv server public URL	Discovered and set by TA
transadv.image.logLevel	Transadv server logging level	info
transadv.resources.requests.memory	Requests memory	1Gi
transadv.resources.requests.cpu	Requests cpu	500m
transadv.resources.limits.memory	Limits memory	8Gi
transadv.resources.limits.cpu	Limits cpu	16000m
transadv.livenessProbe.initialDelaySeconds	Container liveness probe initial delay seconds. Advanced troubleshooting.	60

Parameter	Description	Default
transadv.livenessProbe.timeoutSeconds	Container liveness probe timeout seconds. Advanced troubleshooting.	3
transadv.livenessProbe.periodSeconds	Container liveness probe period seconds. Advanced troubleshooting.	5
transadv.livenessProbe.failureThreshold	Container liveness probe failure threshold. Advanced troubleshooting.	6
transadv.readinessProbe.initialDelaySeconds	Container readiness probe initial delay seconds. Advanced troubleshooting.	60
transadv.readinessProbe.timeoutSeconds	Container readiness probe timeout seconds. Advanced troubleshooting.	3
transadv.readinessProbe.periodSeconds	Container readiness probe period seconds. Advanced troubleshooting.	5
transadv.readinessProbe.failureThreshold	Container readiness probe failure threshold. Advanced troubleshooting.	6
transadvui.image	Transadv UI image tag. Advanced troubleshooting.	
transadvui.imagePullSecret	Image pull secret. Used to access entitled registry. Name must be ibm-entitlement-key. See Planning for more details.	
transadvui.image.logLevel	Transadv UI logging level	info
transadvui.useSecureCookie	Use secure cookie for Transadv UI	true
transadvui.resources.requests.memory	Requests memory	1Gi
transadvui.resources.requests.cpu	Requests cpu	500m

Parameter	Description	Default
transadvui.resources.limits.memory	Limits memory	4Gi
transadvui.resources.limits.cpu	Limits cpu	16000m
transadvui.livenessProbe.initialDelaySeconds	Container liveness probe initial delay seconds. Advanced troubleshooting.	60
transadvui.livenessProbe.timeoutSeconds	Container liveness probe timeout seconds. Advanced troubleshooting.	5
transadvui.livenessProbe.periodSeconds	Container liveness probe period seconds. Advanced troubleshooting.	30
transadvui.livenessProbe.failureThreshold	Container liveness probe failure threshold. Advanced troubleshooting.	6
transadvui.readinessProbe.initialDelaySeconds	Container readiness probe initial delay seconds. Advanced troubleshooting.	5
transadvui.readinessProbe.timeoutSeconds	Container readiness probe timeout seconds. Advanced troubleshooting.	5
transadvui.readinessProbe.periodSeconds	Container readiness probe period seconds. Advanced troubleshooting.	30
transadvui.readinessProbe.failureThreshold	Container readiness probe failure threshold. Advanced troubleshooting.	6

CASE installer configuration

The following options are available for the CASE installer commands:

Options for install action:

```
oc ibm-pak launch \
    $CASE_NAME \
    --version $CASE_VERSION \
    --inventory v2InstallProduct \
    --namespace $TA_PROJECT \
```

```
    --action install \
    --args "[OPTIONS]"

--licenseType <true|false>          : REQUIRED: Must be used and set to a valid
license type.
--acceptLicense <true|false>        : REQUIRED: Must be used and set to true to
proceed with install.
--installIbmCatalog <true|false>    : OPTIONAL: If set to true, the IBM operator
catalog will be installed if it is not already. Default is false.
--secret <secret>                   : OPTIONAL: Specify a secret to use to pull
the Tranformation Advisor images from entitle registry.
--registry <registry>               : OPTIONAL: Specified entitled registry, e.g.
cp.icr.io.
--user <user>                       : OPTIONAL: Specify user to access the
entitled registry.
--pass <password>                   : OPTIONAL: Specify password for user to
access entitled registry.
--persistence <true|false>          : OPTIONAL: If persistence is required for
Transformation Advisor (Default is true).
--accessMode <accessMode>           : OPTIONAL: storage accessMode. Default is
ReadWriteOnce.
--persistenceClaimCouchDB <claim>   : OPTIONAL: Use an existing persistence claim
for CouchDB.
--persistenceClaimNeo4j <claim>     : OPTIONAL: Use an existing persistence claim
for Neo4j.
--storageClass <storage class>      : OPTIONAL: Recommended way use persistence
with Transformation Advisor. Specify a valid storage class to use.
--supplementalGroups [gid,...]      : OPTIONAL: May be used if using file system
based storage to ensure database container has read/write permission for the
storage.
--hostName <hostname>               : OPTIONAL: hostname to access cluster.
Transformation Advisor will discover and set this value. It should not need to be
changed for most environments.
--apiEndpoint <apiEndpoint>         : OPTIONAL: API url for the cluster.
Transformation Advisor will discover and set this value. It should not need to be
changed for most environments.
--authIssuerEndpoint <aiEndpoint>   : OPTIONAL: Auth issuer endpoint for the
cluster. Transformation Advisor will discover and set this value. It should not
need to be changed for most environments.
--publicUrlServer                   : OPTIONAL: See docs for more information.
Transformation Advisor will discover and set this value. It should not need to be
changed for most environments.
--publicUrlUI                       : OPTIONAL: See docs for more information.
Transformation Advisor will discover and set this value. It should not need to be
changed for most environments.
--customCACert <file path>          : OPTIONAL: Specify file to use as custom CA
cert.
--authConfigFile <file path>        : OPTIONAL: Specify file to use to configure
third party authentication.
--namespaceScoped <true|false>      : OPTIONAL: If ommitted, defaults to false.
This will make the operator to be insalled into openshift-operators namespace and
manage all the namespaces, operand will go into namespace specified by --namespace
attribute. If set to true, operator and operand are installed into namespace
specified by --namespace attribute.
--taHelp                            : OPTIONAL: Display options available
```

Options for uninstall action:

```
oc ibm-pak launch \
    $CASE_NAME \
    --version $CASE_VERSION \
    --case ibm-transadv \
    --inventory v2InstallProduct \
```

```
    --namespace $TA_PROJECT \
    --action uninstall \
    --args "[OPTIONS]"
```

```
--uninstallIbmCatalog <true|false>    : OPTIONAL: If set to true, the IBM operator
catalog will be uninstalled. Default is false.
--uninstallTaCatalog <true|false>     : OPTIONAL: If set to true, the IBM operator
catalog will be uninstalled. Default is false.
```

Configure Third Party Authentication

To configure the third party authentication, please follow the headings below:

Update Third Party OAuthClient or OAuthApp

You need to configure the redirection URL to allow the OAuthClient or OAuthApp redirect to TA UI route.

The redirection URL is **`your-ta-ui-route/auth/callback`**

For example, **`https://ta.apps.ken.cp.fyre.ibm.com/auth/callback`**, where **`https://ta.apps.ken.cp.fyre.ibm.com`** is the TA UI route.

You can get this route from OCP UI from the left navigation: **`Networking -> Routes -> ta-ui-route -> Location`**

Client ID and Secret

Before TA 3.4.0, you need to update OAuthClient's or OAuthApp's client ID and secret in the Transformation Advisor instance configuration using the properties **`authentication.oidc.clientId`** and **`authentication.oidc.clientSecret`**.

TA 3.4.0 or after, you need to supply to the client Id and client secret in a secret **`transformation-advisor-secret`** before you install Transformation Advisor. Here is the command:

```
oc create secret generic transformation-advisor-secret \
--from-literal=clientId=your-clientId-value \
--from-literal=clientSecret=your-clientSecret-value
```

Alternatively, You can update your secret after the installation of Transformation Advisor. Here is the command:

```
oc patch secret transformation-advisor-secret \
-p '{"data":{"'"clientId"'": "'"your-clientId-value"'", "'"clientSecret"'":
"'"your-clientSecret-value"'"}}' --type=merge
```

Then, you may need delete the **Server** and **UI** pods, so the new secret values can be applied to the pods.

Note: The **`transformation-advisor-secret`** is used for other internal credentials. Those other credentials are automatically generated, if not already present in the secret, at time the TA instance is created.

Update Egress Network Policy

Make sure you add the endpoints used by the third party to the Egress Network policy. Instruction available at: [Egress Network Policy (ENP)]

Configuring Third Party Authentication - UI Install

IAM

Transformation Advisor can be configured to use IBM Identity and Access Management (IAM) as an authentication source.

1. Perform OpenID Connect (OIDC) registration as per instructions here: https://www.ibm.com/docs/en/cpfs?topic=sign-automated-client-registration-method-3
2. Following the process in step 1, a secret will be created that contains the clientId and clientSecret. Add the clientId and clientSecret to the Transformation Advisor secret (as specified in property: **`authentication.ocp.secretName`**)
3. Update the Transformation Advisor configuration values as follows:

```
      description: "IAM"
      identityRequestEndpoint: "https://cp-console.<OCP domain>:443"
      identityRequestEndpointPath: "/idprovider/v1/auth/authorize"
      identityRequestEndpointScope: "openid+profile+email"
      identityRequestEndpointStatePrefix: ""
      tokenRequestEndpoint: "https://cp-console.<OCP domain>:443"
      tokenRequestEndpointPath: "/idprovider/v1/auth/token"
      tokenVerificationEndpoint: "https://cp-console.<OCP domain>:443"
      tokenVerificationEndpointPath: "/idprovider/v1/auth/userInfo"
```

Github OAuth

The following is an example of the configuration required in the **`thirdparty`** configuration object to use Github OAuth:

```
      description: "github"
      identityRequestEndpoint: "https://github.com"
      identityRequestEndpointPath: "/login/oauth/authorize"
      identityRequestEndpointScope: "openid+offline"
      identityRequestEndpointStatePrefix: ""
      tokenRequestEndpoint: "https://github.com"
      tokenRequestEndpointPath: "/login/oauth/access_token"
      tokenVerificationEndpoint: "https://api.github.com"
      tokenVerificationEndpointPath: "/user"
```

Box OAuth

The following is an example of the configuration required in the **`thirdparty`** configuration object to use Box OAuth:

```
      description: "box"
      identityRequestEndpoint: "https://account.box.com"
      identityRequestEndpointPath: "/api/oauth2/authorize"
      identityRequestEndpointScope: "root_readonly"
      identityRequestEndpointStatePrefix: ""
      tokenRequestEndpoint: "https://api.box.com"
      tokenRequestEndpointPath: "/oauth2/token"
      tokenVerificationEndpoint: "https://api.box.com"
      tokenVerificationEndpointPath: "/2.0/users/me"
```

Configuring Third Party Authentication - CASE Install

Specify a third party authentication configuration file using the **`--authConfigFile`** option with the CASE **`install`** action. The following is an example of that file that uses GitHub OAuth: Do not change the format of the file.

```
# set to true to disable authentication on UI server
TA_AUTH_UI_DISABLED=false
```

```
# set to true to disable authentication on Liberty server
TA_AUTH_LIBERTY_DISABLED=false

# OAuth2 Server client id
TA_AUTH_OIDC_CLIENT_ID=xxx
# OAuth2 Server client secret
TA_AUTH_OIDC_CLIENT_SECRET=xxx

# endpoint to request identity of the OAuth2 Server, no tailing /
TA_AUTH_IDENTITY_REQUEST_ENDPOINT=https://github.com
# path of the endpoint to request identity to OAuth2 Server, with heading /
TA_AUTH_IDENTITY_REQUEST_ENDPOINT_PATH=/login/oauth/authorize
# OAuth2 scope
TA_AUTH_IDENTITY_REQUEST_ENDPOINT_SCOPE=openid+offline
# some OAuth2 state requires minimum length, default to empty
TA_AUTH_CALLBACK_STATE_PREFIX_PADDING=

# endpoint to request token of the OAuth2 Server, no tailing /
TA_AUTH_TOKEN_REQUEST_ENDPOINT=https://github.com
# path of the endpoint to request token of the OAuth2 Server, with heading /
TA_AUTH_TOKEN_REQUEST_ENDPOINT_PATH=/login/oauth/access_token

# endpoint to verify tokens of the OAuth2 Server, no tailing /
TA_AUTH_TOKEN_VERIFICATION_ENDPOINT=https://api.github.com
# path to the endpoint to verify tokens of the OAuth2 Server, with heading /
TA_AUTH_TOKEN_VERIFICATION_ENDPOINT_PATH=/user
```

Enable Bring Your Own Key (BYOK)

You can bring your own certificate (referred as cert in the rest of section) and key used for internal TLS.

Assuming:

1. The public cert is **`public.crt`**, and the private key is **`private.pem`**.
2. Transformation Advisor (TA) is or to be installed in the **`ta`** namespace.

Here is an example of how to get a cert and key pair, and your own cert and key shall be in the same format:

```
openssl req -newkey rsa:2048 -nodes -keyout private.pem -x509 -days 730 -out
public.crt -subj "/C=IE/ST=Cork/L=Cork/O=IBM/CN=internal.ta.ibm.com"
```

After you obtain your own cert and key pair, follow the steps to enable your own cert and key:

1. Switch to **`ta`** namespace, or create one if you haven't installed TA.

   ```
   # switch the project
   oc project ta
   ```

   or

   ```
   # create ta ns if it's not already there
   oc create ns ta
   ```

2. Delete the TA secret **`transformation-advisor-secret`**, if it exists.

   ```
   oc delete secret transformation-advisor-secret
   ```

3. If installing via the CASE installer, pass the **`--customCACert <public.crt>`** where **`<public.crt>`** is the full path to the public.crt file.

If installing via the OpenShift UI, update the **caCert** property in the custom resource YAML. See Installing for more details on accessing the custom resource YAML from the UI.

Here is an example of **caCert** in the custom resource YAML:

```
  tls:
    enabled: true
    caCert: |
      -----BEGIN CERTIFICATE-----
      MIIDKjCCAhICCQCjbqTC95dw+jANBgkqhkiG9w0BAQsFADBXMQswCQYDVQQGEwJJ
      RTENMAsGA1UECAwEQ29yazENMAsGA1UEBwwEQ29yazEMMAoGA1UECgwDSUJNMRww
      GgYDVQQDDBNpbnRlcm5hbC50YS5pYm0uY29tMB4XDTIwMDEyMDEzMjkxMVoXDTIy
      MDExOTEzMjkxMVowVzELMAkGA1UEBhMCSUUxDTALBgNVBAgMBENvcmsxDTALBgNV
      BAcMBENvcmsxDDAKBgNVBAoMA0lCTTEcMBoGA1UEAwwTaW50ZXJuYWwudGEuaWJt
      LmNvbTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKMIGfSptUnimmxZ
      IdVK6uLscJQetel+MX7u4viIaBwdd/IGDE7GHDHEYYFmEfv+gYbVT1+EAkdiLtHG
      SutUMPxpbUyv1xCW+9z3nDInHKzZxHMJJwB5j4+oVq+XkdXzZu2hwuoc0aA7Ek3r
      L6FFPIQi9bcmayuOa7HRpH43+86JuJF8tcx1MrDxWzBJraZNuvDVLs574qr/eM2d
      x5N+qkJqwOy94k0eH+x7kAKRp6vBEcVR+I/HHYDZSnC4UNEX8I/NbCS3wMUXysC9
      lcC2vsIKrCRSn9Fu/ixWwlGy6QV1my4H6ZPtvJV56fcS42523KTDbT628Xa9B3/p
      cX0WZWMCAwEAATANBgkqhkiG9w0BAQsFAAOCAQEADIyp9A4p46DZ6brEbL0e+wWf
      bnnymf1QZWcz4xrrMW2CcKBmRqFIPFPBpSEbCKlsFaZex5863z7dsa5SU7fHRdHF
      Yk9t8mGu2B+yZF6nW4biPmezaDLPi9VUomxcd+/rxWKhZJIufWXxs22AOhNQHeeK
      PSjP8RPCh1Gny7kC3jUz1Q/wd4QF/OGeu+Xf5jhERpJPfjKMtPMPJPGiPYSqhYQM
      VA6G83nZVlPDtnFJ28AzZU2/YtvCzhU66Ua5PjbSG1w6QsXZt/lE3E9utcJ+MNQi
      3JwrLp4/97cupXsGmPPmkvH50LB8ex/N/ra6QOLGLm0gU218yXu7KezOfZNkTw==
      -----END CERTIFICATE-----
```

You can manually copy and paste to the location. Remember the indentation is important. There are 2 extra spaces for each line of the cert than the line of **caCert**.

4. Re-create the TA secret

```
# create key.p12
openssl pkcs12 -export -inkey private.pem -in public.crt -name default -out
key.p12 -passout pass:plain-text-password

# base64 encode private.pem and public.crt
base64 -w 0 ./private.pem > private-base64
base64 -w 0 ./public.crt > public-base64
# on Mac
# base64 ./private.pem > private-base64
# base64 ./public.crt > public-base64

# create key and initial vector for AES-CBC-256 (P)
# key length for aes256 is 256 bits around 32 characters
TA_TEMP_KEY=`LC_CTYPE=C tr -dc A-Za-z0-9_ < /dev/urandom | head -c 32 |
xargs`
# key length for aes256 is 128 bits around 16 characters
TA_TEMP_IV=`LC_CTYPE=C tr -dc A-Za-z0-9_ < /dev/urandom | head -c 16 | xargs`

# create transformation-advisor-secret
oc create secret generic transformation-advisor-secret \
--from-literal=db_username='plain-text-username' --from-
literal=secret='plain-text-password' \
--from-file=ta_public_key=./public-base64 --from-
file=ta_private_key=./private-base64 \
--from-literal=ta_aes_key=$TA_TEMP_KEY --from-literal=ta_aes_iv=$TA_TEMP_IV \
--from-file=key.p12=key.p12
```

You can now proceed with your installation.

Note: The **transformation-advisor-secret** is used for other internal credentials. Those other credentials are automatically generated, if not already present in the secret, at time the TA instance is created.

Create Image Pull Secret to pull Entitled Registry images

To avail of support for IBM Transformation Advisor one needs to provide access permissions to Entitled Registry - this is done via creation of an Image Pull Secret and pointing to it on install. The name of the secret must be **ibm-entitlement-key**. Secret needs to be created in the same namespace where the product instance is installed or globally.
Please refer to the Image Registry Images Access document for more details on that.
Here is how to point to the secret from Transformation Advisor instance configuration page:

```
couchdb:
  imagePullSecret: ibm-entitlement-key

neo4j:
  imagePullSecret: ibm-entitlement-key

transadv:
  imagePullSecret: ibm-entitlement-key

transadvui:
  imagePullSecret: ibm-entitlement-key
```

Provide your own credentials to access TA's internal DB's

IBM Transformation Advisor creates random credentials to communicate with internal DB's (CouchDB and Neo4j) when it gets installed first time. It is possible, however, to provide your own credentials for that purpose. The credentials are kept in a kubernetes **secret** (called **transformation-advisor-secret** by default) object which can be created in the following way:

```
TA_TEMP_KEY=`LC_CTYPE=C tr -dc A-Za-z0-9_ < /dev/urandom | head -c 32 | xargs`
TA_TEMP_IV=`LC_CTYPE=C tr -dc A-Za-z0-9_ < /dev/urandom | head -c 16 | xargs`

oc -n <YOUR_TA_INSTANCE_NAMESPASE> create secret generic <YOUR_SECRET_NAME> \
  --from-literal=db_username=<YOUR_COUCHDB_USERNAME> \
  --from-literal=secret=<YOUR_COUCHDB_PASSWORD> \
  --from-literal=db_nonadmin_user=<YOUR_COUCHDB_NONADMIN_USERNAME> \
  --from-literal=db_nonadmin_secret=<YOUR_COUCHDB_NONADMIN_PASSWORD> \
  --from-literal=neo4j_username=<YOUR_NEO4J_USERNAME> \
  --from-literal=neo4j_secret=<YOUR_NEO4J_PASSWORD> \
  --from-literal=neo4j_auth=neo4j/<YOUR_NEO4J_PASSWORD>
```

Please read the **Re-create the TA secret** section of this doc to see how to create the **public-base64** and **private-base64** files.
You can provide your secret name at installation time of a TA instance in here:
.authentication.ocp.secretName

Edit Cipher Suites and TLS Versions

Cipher Suites are set automatically for Neo4j and UI container.
For the Server and Couchdb container you can set them manually by modifying those variables at install time: **couchdb.security.cipherSuites** and **transadv.security.cipherSuites**. Look up the default values in the table above. Add your own as a whitespace separated string.

Similarly TLS Versions can be set manually for the Couchdb container. The variable for that is **`couchdb.security.tlsVersions`**. See table above for default values.

.

Adding your organisations text and icon to TA landing page

You may if you wish customize the TA landing page by changing the title and/or replacing the TA icon with your own.

Adding additional text to the title

Use **`custom.titleText`** to add your text in front of the **IBM Cloud Transformation Advisor** title on the UI landing page.

Replacing the TA icon

Use the **`custom.landingIcon`** variable to replace TA icon on the landing screen. It consists of four pages. Use the following variables to configure each page individually: **`custom.landingIcon, custom.landingIcon2, custom.landingIcon3, custom.landingIcon4`**.
Provide a URI or a string encrypted image (max 128KiB in length). For example:

```
landingIcon: "<icon URI>"
or
landingIcon: "data:[<mime type>][;charset=<charset>][;base64], <encoded icon>"
```

.

Getting started in the IBM Cloud Transformation Advisor UI

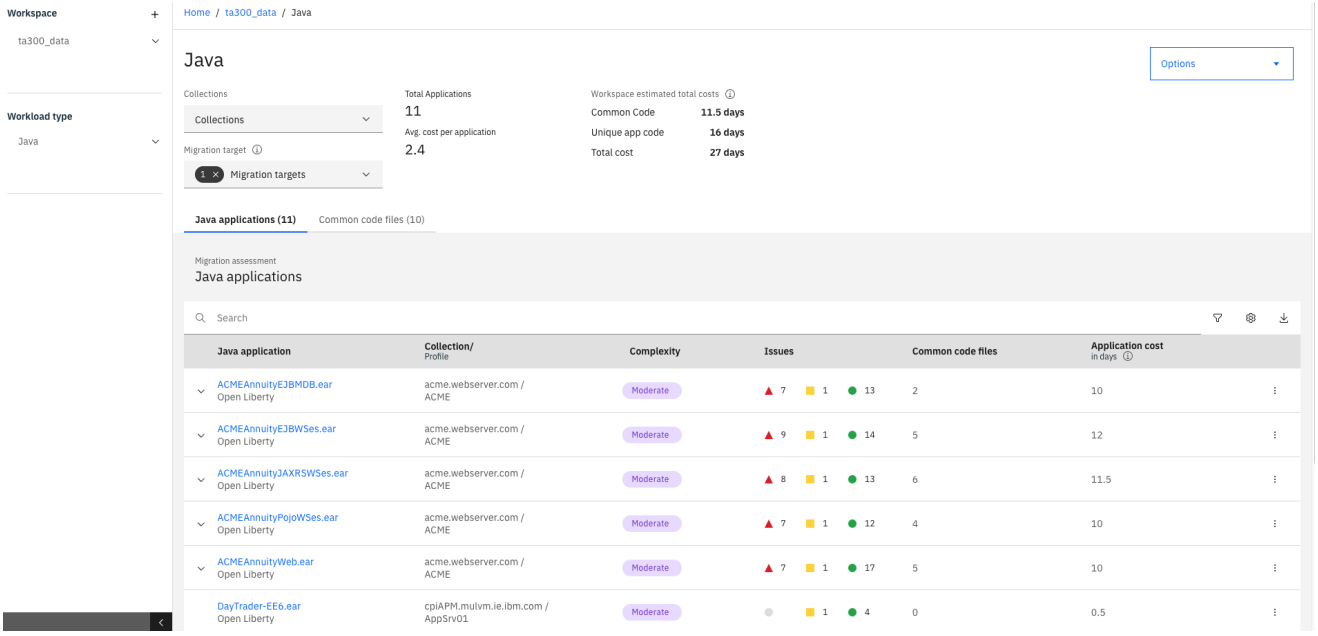
After you have installed IBM Cloud Transformation Advisor, you can access the URL from your browser, as described in the installation instructions. You will land on the Welcome screen for the UI. To get started, complete the following steps:

1. Create and name a new workspace to house your analysis results. You can name your workspace anything you want, such as the name or your project or the portfolio of applications you will be analyzing. A best practice is to choose a name that will help you easily identify your work when you return at a later date.

2. Now you need to add your analysis data to the workspace. Click the 'Download' button to get started with the data collector.

Viewing the analysis results

After the data collector has completed the scan and uploaded data about your applications, you will see a table that displays the results of the analysis for each application in your workspace.



The following details are included in the workspace summary:

- **Total Applications**: The total number of applications in the workspace.
- **Avg. cost per application**: The average number of days of development effort required to migrate an application.
- **Common Code**: The total cost to migrate all the common code in the workspace to the target platform.
- **Unique app code**: The total cost to migrate all the unique app code (code that is not shared between applications) in the workspace to the target platform.
- **Total cost**: The total cost to migrate all the applications and common code in the workspace to the target platform.

The following details are included in the summary table (this is the per-application view):

- **Application Name**: The name of the EAR/WAR file found on the application server.
- **Collection/Profile**: Collection represents the hostname of the machine where the application resides. The profile represents the profile name in the application server where the application is installed.
- **Complexity**: Indicates how complex Transformation Advisor considers this application to be if you were to migrate it to the cloud.
- **Issues**: The number and severity of potential issues with the migration of the application.
- **Common code files**: The number of common code files this application uses. A file is considered common if it is from a Shared Library, or is used by at least one other application.
- **Application cost in days**: Provides an estimate in days for the development effort to perform the migration for just this application. Cost estimates calculated by Transformation Advisor are high-level estimates only and may vary widely based on skills and other factors not considered by the tool.
- **Overflow menu**: Provides the ability to access the migration plan page for each application.

Each column in the table is sortable and you can filter the view by complexity. To see a breakdown of shared library complexity and shared files that are used by an application, you can expand the view. This is helpful in understanding the complete analysis that is applied to each application.

To see the analysis details for any application, click on the row. The complexity rating is explained and more information is provided about several different aspects of the analysis. At the end of the page, there are links to three reports which go into even more detail about the analysis.

Understanding Costs

There are a number of things to understand about the costs presented by Transformation Advisor:

- All costs are listed in days and represent the development effort required to complete the migration.
- They do not include costs of testing the new deployment and full configuration for production.
- The costs at the workspace level assume the same team is migrating the whole workspace and will be faster at resolving issues that repeat across applications.
- The costs in the summary table assume the migration of that application in isolation.
- This means that if you add up the cost for each application in the summary table it will be greater than the total cost shown for the workspace. This is because it does not take into account:
    - the fact that common code needs to be migrated only once per workspace (not once per application).
    - the savings offered by the same issues repeating across applications.

Migration Targets

Transformation Advisor will always perform an analysis for migrating to Open Liberty, WebSphere Liberty and WebSphere traditional (for migration to WebSphere base in containers). You can use the **Migration target** dropdown on the migration page to change the analysis results that are displayed in the recommendations table.

Technology Report

The Application Technology Evaluation Report identifies the editions of WebSphere Application Server that are best suited to run the application. The report provides a list of Java EE programming models that are used by the application and indicates which platforms will support the application.

Application Technology Evaluation Report

07/10/19 09:17
/opt/IBM/WebSphere/AppServer/profiles/Dmgr01/config/cells/transAdvisor1Cell01/applications/DayTrader-EE6.ear/DayTrader-EE6.ear

Scan options: --baseEdition --coreEdition --libertyBuildpackEdition --ndEdition --zosEdition --traditional --liberty --excludePackages=com.ibm, com.informix, com.mchange, com.microsoft, com.sybase, com.sun, java, javax, net, oracle, org, sql), _ibmjsp

WebSphere Application Server V9.0

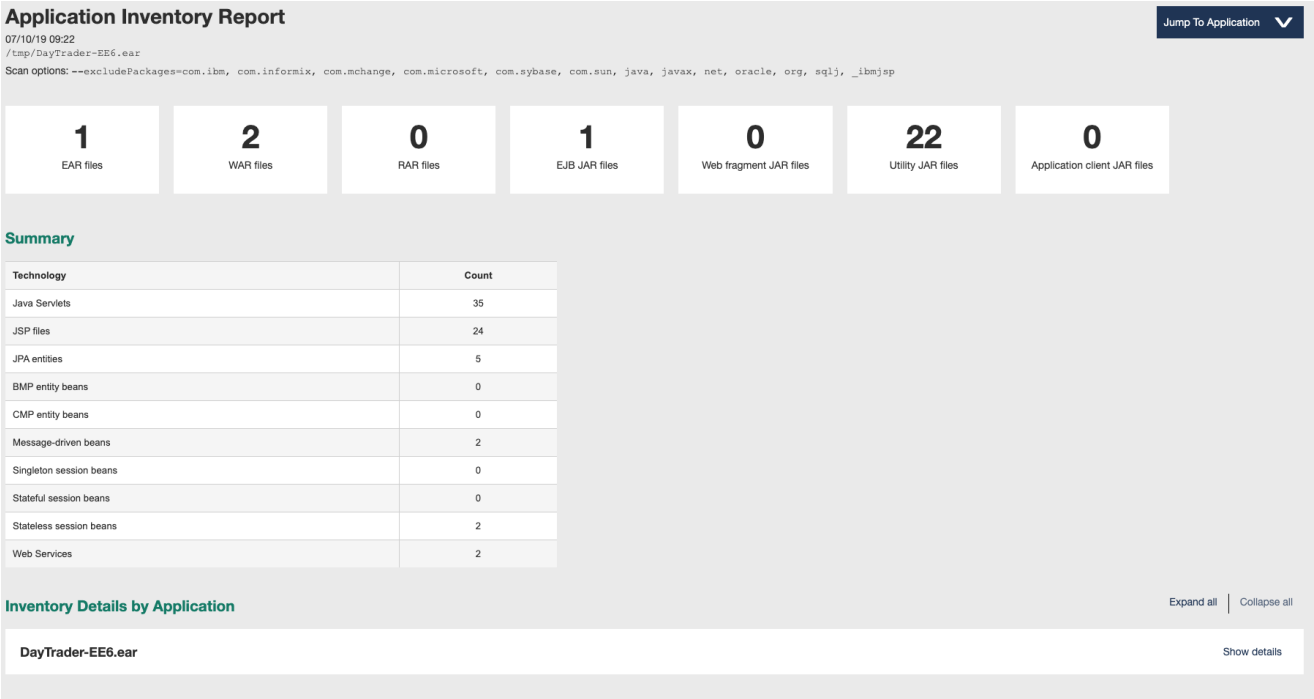
The highlighted columns indicate which IBM platforms fully support the technologies used by the included application.

Recommendation: Detailed migration analysis should be used to determine if there are migration issues that must be addressed before deploying your application.

	Liberty for Java on IBM Cloud	Liberty Core	Liberty	WebSphere traditional	Network Deployment Liberty	Network Deployment traditional	Liberty for z/OS	WebSphere traditional for z/OS
WEB SERVICES TECHNOLOGIES								
Java API for RESTful Web Services (JAX-RS)	✓	✓	✓	✓	✓	✓	✓	✓
WEB APPLICATION TECHNOLOGIES								
Java Servlet	✓	✓	✓	✓	✓	✓	✓	✓
JavaServer Faces (JSF)	✓	✓	✓	✓	✓	✓	✓	✓
JavaServer Pages / Expression Language (JSP/EL)	✓	✓	✓	✓	✓	✓	✓	✓
Standard Tag Library for JavaServer Pages (JSTL)	✓	✓	✓	✓	✓	✓	✓	✓
ENTERPRISE APPLICATION TECHNOLOGIES								
Enterprise JavaBeans (EJB) Lite subset	✓	✓	✓	✓	✓	✓	✓	✓
Message-Driven Beans (MDB)	✓		✓	✓	✓	✓	✓	✓
Java Persistence (JPA)	✓	✓	✓	✓	✓	✓	✓	✓
Common Annotations for the Java Platform	✓	✓	✓	✓	✓	✓	✓	✓
Java Message Service (JMS) API	✓		✓	✓	✓	✓	✓	✓
Java Transaction API (JTA)	✓	✓	✓	✓	✓	✓	✓	✓
JAVA EE-RELATED SPECIFICATIONS IN JAVA SE								
Java Architecture for XML Binding (JAXB)	✓	✓	✓	✓	✓	✓	✓	✓
Java Database Connectivity (JDBC)	✓	✓	✓	✓	✓	✓	✓	✓

Inventory Report

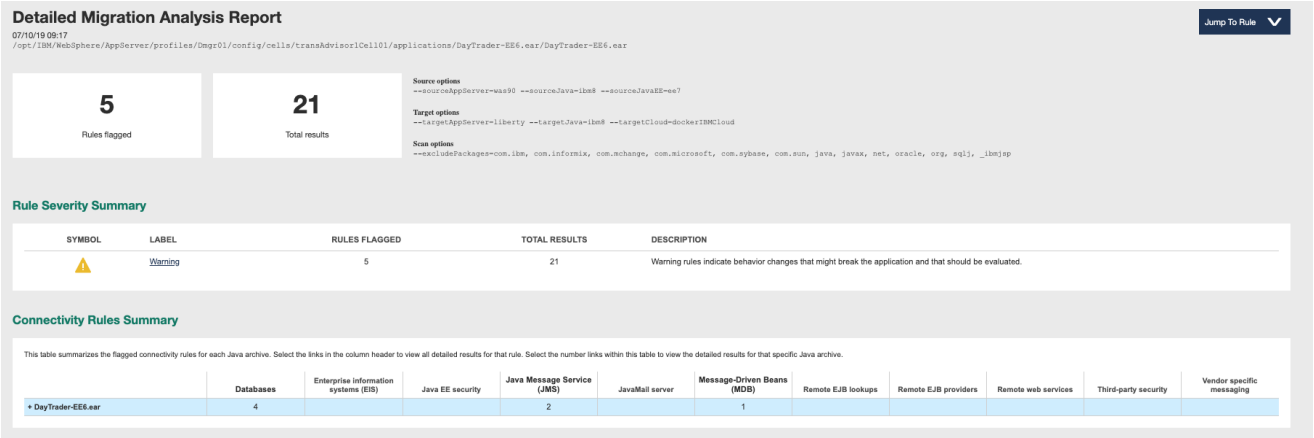
The Application Inventory Report helps you examine what is in your application, including the number of modules and the technologies in those modules. It also gives you a view of all the utility JAR files in the application that tend to accumulate over time. Potential deployment problems and performance considerations are also covered.



Analysis Report

The Detailed Migration Analysis Report does a deep dive on the preferred migration target to help you understand any migration issues, like deprecated or removed APIs, Java SE version differences, and Java EE behavioural differences. Note that Transformation Advisor uses a rule system based on commonly occurring events that are seen in real applications to enhance the base reports and provide practical guidance. As a result, some items may show a different severity level in Transformation Advisor than they do in the detailed binary scanner reports.

Note: The analysis report may specify Open Liberty as the target even if you have selected WebSphere Liberty as your target. This happens if the analysis for both of these targets are the same, so the report is still valid.



Preparing for your migration

After you decide to migrate an application with a preferred migration path, open the overflow menu and select the **View migration plan** option to go to the Migration plan page.

Transformation Advisor automatically generates a migration bundle with the artifacts you will need to containerize your application running on Liberty and deploy it to OpenShift Cloud Platform. For binary applications, you can upload any dependencies and external drivers that were detected. When you deploy, the bundle is sent to the Git repository that you specify. You can also download the bundle.

.

How to collect data

There are three approaches that you can take to collect data for Transformation Advsior

- Using the data collector
- Using the built-in WSADMIN command for WebSphere Application Server
- Using the Migration Toolkit for Application Binaries (aka: the Binary Scanner)

Using the data collector

The data collector is a tool that gathers information about middleware deployments in your environment to help IBM Cloud Transformation Advisor provide you with a migration analysis of Java™ EE applications running on IBM WebSphere Application Server, Apache Tomcat, Red Hat JBoss, or Oracle WebLogic application servers. The tool generates one .zip folder per profile/domain and places analysis results within that directory.

For Java applications, the tool leverages the Migration Toolkit for Application Binaries. You may use the Migration Toolkit for Application Binaries (AKA the "binary scanner") directly. This may be particularly useful in some troubleshooting scenarios. For information on how to download and use the Migration Toolkit for Application Binaries please click here. If these tools are already in use, then the same security requirements which are required to run them are also required to run the data collector.

The data collector will collect configuration information on WebSphere Application Server installations at Version 6.1 or later.
The exact configuration information that is collected can be found here

To analyze your applications, the data collector requires the following access:

- Read access to the application server installation directory and all subdirectories
- Read access to profile directories
- File and directory creation and write access to the current directory

The data collector does not gather any data related to the following:

- Message content or data processed by workload
- Logs or log data
- Password information

Downloading the data collector

To get started, complete the following steps from the Transformation Advisor UI:

1. Create a workspace.
2. On the new workspace page, click the **data collector** button and follow the instructions to download.

Installing the data collector

To install the data collector, log on to your application server with the application owner's user credentials and complete the following steps:

1. Copy the downloaded file to your application system in a directory where it has read-write-execute access.
2. Decompress the downloaded file by issuing the command for your operating system:

Linux: **tar xvfz transformationadvisor-Linux_<WORKSPACE_NAME>.tgz**

AIX: **gunzip -c transformationadvisor-AIX_<WORKSPACE_NAME>.tgz | tar xf -**

Solaris: **tar xvfz transformationadvisor-Solaris_<WORKSPACE_NAME>.tgz**

z/OS: **gunzip -c transformationadvisor-zOS_<WORKSPACE_NAME>.tgz | tar xf -**

Windows: **unzip transformationadvisor-Windows_<WORKSPACE_NAME>.zip**

3. Go to the data collector directory:

```
cd transformationadvisor*
```

Required resources

System	Memory (GB)	CPU (cores)	Disk space (GB)
Data Collector	2	2	0.5

Running the data collector

When you run the data collector for the first time, the license will be displayed. Accept the terms to continue.

WARNING: The data collector should not be run on production servers.

To analyze both your applications and their configuration, run the command for your domain:

IBM WebSphere: **./bin/transformationadvisor -w <WEBSPHERE_HOME_DIR> -p <PROFILE_NAME> [--ignore-missing-binary --ignore-missing-shared-library --applications --applications-file --skip-applications --skip-applications-file --no-upload] ([] denotes optional arguments)**

Oracle WebLogic: **./bin/transformationadvisor --web-logic-config-file <Path to the WebLogic domain config.xml file> [--applications --applications-file --skip-applications --skip-applications-file --no-upload] ([] denotes optional arguments)**

JBoss: **./bin/transformationadvisor --jboss-config-dir <Path to the JBoss server configuration directory> [--applications --applications-file --skip-applications --skip-applications-file --no-upload] ([] denotes optional arguments)**

Apache Tomcat: **./bin/transformationadvisor --tomcat-home-dir <TOMCAT_HOME_DIR> --tomcat-config-dir <TOMCAT_CONFIG_DIR> [--applications --applications-file --skip-**

```
applications --skip-applications-file --no-upload] ([] denotes optional arguments)
```

To view command line options that are available for the data collector, use the --help option.

Viewing your data

Depending on the number, size, and complexity of your applications, the data collector may take some time to execute and upload the results. During this process, you can keep track of its progress by checking your command line.

If there is a connection between your system and your new collection, the data collector automatically uploads the results to Transformation Advisor. A detailed analysis that includes several reports is provided to help you understand issues and where code changes might be required.

If there is no connection, the data collector will return a .zip file containing your application data. You will need to manually upload the zip file using the **Upload data** button on the Workspace page.

Transformation Advisor data collector and Java

The data collector requires version 1.7+ to run. In the first instance, the data collector will look for a compatible version of Java that is used by WebSphere (if this is a WebSphere collection). If it cannot find the WebSphere Java version, or if the WebSphere version is not compatible (i.e. < 1.7), or this is not a WebSphere collection, then the data collector will attempt to use the Java specified in the **`--java-home`** data collector argument. If the **`–java-home argument`** is not provided, it will use the Java specified with the **`JAVA_HOME`** environment variable. If none of those options result in finding a compatible version of Java, the data collector will use the Java that is packaged with the data collector.

Replacing version of Java in the data collector

In some circumstances, a compatible version of Java may not pre-exist on the system that you want to run the data collector on, and the Java that is packaged with the data collector is also not compatible with the system on which it is being run. For example, this can arise with the Solaris data collector which contains Java for AMD64 architecture. If your Solaris is running on SPARC, then the packaged Java will not work. In these cases, you can build your own version of the data collector, with an appropriate Java version.

Take the following steps:

- Download a version of that data collector via the Transformation Advisor UI
- Unpack the compressed data collector
- Locate the “jre” folder in the unpacked data collector and replace it with a jre from a desired Java version
- *OPTIONAL:* There is a file in the unpacked data collector called uploadEndpoint.json. This contains the location of the Transformation Advisor server, and a unique key for uploading to a specific workspace. You can remove the uploadEndpoint.json file. This means that the data collector will not be able to automatically upload the collection to Transformation Advisor, but rather will produce a zip that you can manually upload using the Transformation Advisor UI. Removing the uploadEndpoint.json has the advantage of allowing you to create a custom data collector that can be readily copied between environments. In effect, it breaks the link between the data collector and a specific workspace in Transformation Advisor.
- Compress the data collector

The compressed data collector can now be copied to whatever environment you wish to perform the collection on.

Migrating From SunOS and Solaris architectures

Transformation Advisor provides a data collector for SunOS/Solaris environments. It has been tested on SunOS 5.10, 5.11 (Solaris 10, 11), but may work on older versions. The data collector provided for SunOS/Solaris packages a JRE for use on SunOS/Solaris on AMD64 architecture. If you want to run the data collector on SunOS/Solaris SPARC architecture, then you need to have a compatible version of Java available on the System where you are running the data collector.

Known Issues:

- Before running the data collector on SunOS/Solaris, please check the version of bash on the environment. If the version is < 4.x, please upgrade to 4.x to ensure correct functioning of the data collector.

Migrating from old versions of WebSphere

The Transformation Advisor data collector supports collecting from WebSphere v6.1+. For WebSphere migrations from versions older than 6.1, you need to manually find all of the application binaries deployed on the system and put them into some single location. You can then run the data collector using the “-o” option to point to that location and perform an analysis on the applications. When you run the data collector with the -o options, it will ask you to choose the source WebSphere and Java version. If it does not show your exact versions, select the values that are closest to your actual versions to get the most accurate results. In this situation, the analysis may not catch all issues, issues from version of WebSphere and Java older than the supported rules will not be flagged.

Migrating from supported application servers

Transformation Advisor supports collection from the following application servers:

- WebSphere
- Red Hat JBoss
- Weblogic
- Tomcat.

Migrating from application servers that are not supported

If you want to run an analysis on applications from application servers that are not supported (e.g. Glassfish), you can do the following:

- Manually collect the application binaries from the system, and place in a directory.
- Run the data collector and point it at the applications location using the “-c” option. This runs a base set of Java rules plus some Tomcat specific rules. The Tomcat specific rules may or may not apply to the application server that you are analyzing, and need further investigation. In future versions of the data collector, there will be the option to just run the generally applicable Java rules. The Tomcat rules that need further investigation are as follows:

```
Set the sharing scope on resource references
Spring applications might fail to run from a non-expanded WAR file
Stub classes must be included when using remote Enterprise JavaBeans (EJB) 2.x
The getRealPath method previously returned null for files that do not exist
The OSGI remote bundle repository service API is unavailable
The OSGI Remote Service Admin API is unavailable
Transaction propagation is not supported for Enterprise JavaBeans (EJB) remote
```

```
interfaces
Use correct case for tag attribute names
Use Java EE deployment descriptors and WebSphere bindings to define resource link
references
Use Java EE deployment descriptors and WebSphere bindings to define resource
references
Use Java EE deployment descriptors to define context lifecycle listeners
Use Java EE deployment descriptors to define context parameters
Use Java EE deployment descriptors to define environment references
Use Java EE deployment descriptors to define missing security roles
Validate the result of concatenation with getRealPath("")
Validate the result of concatenation with getRealPath("/")
Web Services Notification (WS-Notification) is unavailable
```

Customize the scan options used to generate report files

The customCmd.properties file under the /conf directory is used to config the scan options used to generate the report file during scanning the application. User can edit this file to customize the scan options.

Defining your own rules

Since the release of Transformation Advisor v2.5 you can define your own custom analysis rules to detect scenarios specific to your application migration. The user-defined rules can be easily created as described in detail here. The results will display in Transformation Advisor UI in the same manner as the pre-written rules that come out of the box.

data collector command line tool options

General options:

--version

Displays the version of the data collector.

--help

Displays all options for the data collector.

--java-home

Specifies the version of Java used for the data collector.

-J-X

JVM options that are passed to the data collector. -J will be stripped and -X and remaining options will be passed to the Java runtime. For example, **`-J-Xmx4G`** will set the max heap size to 4GB for the data collector runtime.

--no-upload

Flag to indicate that no upload to Transformation Advisor server will take place once the collection completes

Options for IBM WebSphere:

-w, --was-home

The location of the WebSphere installation directory.

-p, --profile-config

If provided, the data collector will connect to the WebSphere JVM associated with that profile and collect the configuration information. If you have multiple profiles within your WebSphere installation directory, you can supply this option multiple times. The syntax for this option is: **`--profile-config <Profile Name>`**

Example: **`--profile-config AppSrv01  --profile-config Dmgr01`**

-a, --applications

By default, the data collector collects configuration information for all applications deployed on a profile. If this option is specified, the tool will only collect configuration information for the applications that are listed. The syntax to specify multiple applications is: **`--applications <application_1 Name>... <application_n name>`**

Example: **`--applications app1 app2 app3`**

-f, --applications-file

Similar to the --applications option, this option allows the data collector to obtain application names from the provided file, and only collect configuration data for applications that are defined in that file.

Example: **`--applications-file /tmp/applicationsToScan.txt`**

The file can be a comma separated and/or line separated list of applications to include, e.g.

```
app1.ear,app2.ear
app3.ear
```

-s, --skip-applications

By default, the data collector collects configuration information for all applications deployed on a profile. If this option is specified, the tool will skip the applications that are listed. The syntax to specify multiple applications is: **`--skip-applications <application_1 Name>...<application_n name>`**

Example: **`--skip-applications app1 app2 app3`**

--skip-applications-file

Similar to the --skip-applications option, this option allows the data collector to obtain application names from the provided file, and skip those applications that are defined in that file.

Example: **`--skip-applications-file /tmp/applicationsToScan.txt`**

The file can be a comma separated and/or line separated list of applications to skip, e.g.

```
app1.ear,app2.ear
app3.ear
```

--ignore-missing-binary

Signals the scan to skip applications that do not have binary files.

-i, --ignore-missing-shared-library

Signals the scan to run on applications that have a nonexistent shared library.

--scan-binary-location

Signals the scan to run against the expanded directory of the deployed application. By default, the data collector will scan the binary file which is uploaded to the WAS server during deployment time. If this option is specified, the data collector will scan the expanded directory of the deployed application on the WAS server.

-s, --scan-node

Signals the scan to run for managed profiles on the node.

-o, --outside-location

Location of a directory outside of the WAS home location that contains application binary files that are to be scanned. The syntax is: **./transformationadvisor -o <location of applications Outside WAS>**

Note: If this option is specified, all previous arguments are ignored. The data collector will ask you to select the WebSphere version number and Java version.

Options for Oracle WebLogic:

-l, --web-logic-config-file

Path of the WebLogic domain config.xml file. If you have multiple domains, you can specify this option multiple times. The syntax is: **--web-logic-config-file <Path of the config.xml file>**

Example: **--web-logic-config-file /home/oracle/Oracle/Middleware/Oracle_Home/user_projects/domains/base_domain/config/config.xml**

-g, --web-logic-apps-location

Location of the directory that contains WebLogic application binary files. The syntax is: **--web-logic-apps-location <WebLogicapps location>**

Note: If this option is specified, all previous arguments are ignored. The data collector will ask you to select the Java version used for the Weblogic server.

Options for JBoss:

-j, --jboss-config-dir

Path of the JBoss server configuration directory. If you have multiple servers, you can specify this option multiple times. The syntax is: **--jboss-config-dir <Path of server configuration directory>**

Example: **--jboss-config-dir /root/EAP-7.1.0/standalone/configuration**

-b, --jboss-apps-location

Location of the directory that contains JBoss application binary files. The syntax is: **--jboss-apps-location <JBoss applications location>**

Note: If this option is specified, all previous arguments are ignored. The data collector will ask you to select the Java version used for the JBoss server.

Options for Apache Tomcat:

-t, --tomcat-home-dir

Path of the Tomcat CATALINA_HOME directory. If this argument is specified and the **--tomcat-config-dir** argument is not, then it is assumed that **--tomcat-config-dir** is the same as **--tomcat-home-dir**. If one or more **--tomcat-config-dir** options are specified in addition to **--tomcat-home-dir**, the **--tomcat-home-dir** will not be treated as a config directory, unless it's explicitly listed in one of the **--tomcat-config-dir** options. The syntax is: **`--tomcat-home-dir <Directory of Tomcat home>`**

-d, --tomcat-config-dir

Path of the Tomcat CATALINA_BASE directory. The **--tomcat-home-dir** (CATALINA_HOME) must be specified in addition to this option. If you have multiple CATALINA_BASE directories for a multi-instance Tomcat installation, you can supply this option multiple times. The syntax is: **`--tomcat-config-dir <Directory of Tomcat Configuration>`**

-c, --tomcat-apps-location

Location of the directory that contains Tomcat application binary files. The syntax is: **`--tomcat-apps-location <Tomcat appslocation>`**

Note: If this option is specified, all previous arguments are ignored. The data collector will ask you to select the Java version used for the Tomcat server.

Using the built-in WSADMIN command for WebSphere Application Server

The wsadmin command available with WebSphere has a built-in option for scanning your installed applications and generating a set of zip files that can be uploaded directly into Transformation Advisor. The availability of this command depends on the WebSphere version and Fix Pack you are running:

- WebSphere 8: Version 8.5.5.23 or later
- WebSphere 9: Version 9.0.5.14 or later

The full details of this and all instructions can be found here

Using the Migration Toolkit for Application Binaries

An alternative to using the Transformation Advisor Data Collector is to use the Migration Toolkit for Application Binaries to generate a data collection that can be uploaded to IBM Cloud Transformation Advisor. Details of how to do this are outlined here.

.

Controlling collection names

When data is uploaded Transformation Advisor automatically detects the host machine the data came from (the application server where the apps were running) and creates a corresponding collection and adds the applications there. In some cases the host name may not be meaningful, or due to the nature of virtualization all your different application servers may appear to have the same host name - so we provide a number of options to let you specify the collection name on upload.

Controlling the collection name using the data collector

You have two options for controlling the name of the collection when using the data collector

1. Add the param `--collection-name <arg>` where `<arg>` is the name of the collection you want, when you execute the data collector.
2. Edit the file `uploadEndpoint.json` in your data collector and add `?collectionName=<arg>` to the attribute `uploadEndpoint` where `<arg>` is the name of the collection you want

Controlling the collection name using the filename

Transformation Advisor accepts zip files produced by the data collector for upload.
You can embed the name of the collection in the zip file name by adding
`__collectionName_<collectionName>` to the file name.
For example you have a zip called AppSrv01.zip
To assign it the collection name 'happy' on upload then you can change the name of the file to
`AppSrv01__collectionName_happy.zip`

Controlling the collection name using the UI

When uploading data via the UI the default option is to automatically detect the collection name. You can also choose to

- Select and existing collection
- Enter a collection name

.

Collecting on WebSphere Liberty or Open Liberty

Note: The document will refer to 'Liberty' and this refers to both WebSphere Liberty and Open Liberty

The Data Collector is designed to analyze the WebSphere Application Server traditional and similar runtimes. However, it also provides tools to assist in modernizing from Liberty (typically on a VM) to Liberty in Containers.
To do this we will use the binary scanner that is included with the Data Collector, or the binary scanner can be downloaded separately.
The results can be reviewed locally and will not be uploaded into Transformation Advisor.

Prerequisite

- A copy of the application war or ear file(s) that are to be modernized to Liberty in Containers
- A copy of the Liberty Migration Bundle

Downloading the scanning tool

If you have Transformation Advisor installed you can download the Data Collector and extract the binary scanner, otherwise you should download the binary scanner directly.

Downloading and installing the Data Collector (assumes you have Transformation Advisor installed)

To get started, complete the following steps from the Transformation Advisor UI:

1. Create a workspace.
2. On the new workspace page, click the **data collector** button and follow the instructions to download.
3. Copy the downloaded file to your system in a directory where it has read-write-execute access.
4. Decompress the downloaded file by issuing the command for your operating system:
   Linux: `tar xvfz transformationadvisor-Linux_<WORKSPACE_NAME>.tgz`
   AIX: `gunzip -c transformationadvisor-AIX_<WORKSPACE_NAME>.tgz | tar xf -`
   Solaris: `tar xvfz transformationadvisor-Solaris_<WORKSPACE_NAME>.tgz`
   z/OS: `gunzip -c transformationadvisor-zOS_<WORKSPACE_NAME>.tgz | tar xf -`
   Windows: `unzip transformationadvisor-Windows_<WORKSPACE_NAME>.zip`
5. Go to the data collector lib directory: `cd transformationadvisor*/lib`
6. Copy the binary scanner jar file to the location where you have your applications `cp ta.binaryAppScanner* binaryAppScanner.jar`

Downloading and installing the binary scanner

To get started, complete the following steps:

1. Go to the Migration Toolkit for Application Binaries download page: https://www.ibm.com/support/pages/migration-toolkit-application-binaries
2. Download the Migration Toolkit for Application Binaries
3. Run the following command `java -jar binaryAppScannerInstaller.jar`
4. Accept the License and use the default values for product location when prompted
5. Copy the binary scanner to your application location `cp binaryAppScanner.jar binaryAppScanner.jar`

Running the binary scanner to modernize to Liberty in Containers

You will execute the binary scanner against your application binaries, review the results and combine your existing server.xml file with the Liberty Migration Bundle to complete the modernization to Liberty in Containers.

The steps are as follows:

1. Run the binary scanner analyze option against your application binary, replacing the APP_NAME with the name of your application

```
java -jar binaryAppScanner.jar <APP_NAME.war> --analyze --
sourceAppServer=liberty --sourceJava=ibm8 --targetCloud=containers
```

2. The analysis report will be generated:
   - **APP_NAME.war_AnalysisReport.html**: A html file that shows all the potential modernization issues for the WebSphere Liberty runtime
   - OPTIONAL: You can generate additional reports with the following commands
     - `java -jar binaryAppScanner.jar <APP_NAME.war> --inventory`
       - Generate the inventory report
     - `java -jar binaryAppScanner.jar <APP_NAME.war> --evaluate`
       - Generate the technology evaluation report
3. Run the binary scanner generate configuration option against your application binary, replacing the APP_NAME with the name of your application

   ```
   java -jar binaryAppScanner.jar <APP_NAME.war> --generateConfig
   ```
4. The **server.xml** file will be generated:
   - **APP_NAME.war_server.xml**: A server.xml file that includes the list of features required for the application
5. Review the analysis file, which will be called **APP_NAME.war_AnalysisReport.html**
   - NOTE: *We expect the issues to be Informational and that you will not need to make any code changes before your application will run on Liberty in Containers. If there are issues that are not Informational (for example if you are also changing Java version) some code changes may be necessary*
6. Download the Liberty Migration Bundle
7. Update the Liberty Migration Bundle for your chosen runtime and application
   1. Unzip the Liberty Migration Bundle
   2. Delete the Dockerfile for the runtime you are **not** modernizing to.
   3. Rename that Dockerfile for the chosen runtime from Dockerfile.*runtime* to Dockerfile
   4. Delete the application-cr.yaml file for the runtime you are **not** modernizing to.
      - The file can be found here in the **deploy** directory
   5. Rename that application-cr.yaml for the chosen runtime from application-cr.yaml.*runtime* to application-cr.yaml
   6. Edit the application-cr.yaml file and replace APP_NAME with your application name.
   7. Rename the file target/APP_NAME.war.placeholder, replacing APP_NAME with your application name
   8. In the migration bundle, replace the provided **server.xml** file with your existing **server.xml** file for your application
      - The file can be found here: **src/main/liberty/config/server.xml**
   9. In the migration bundle, replace the featurelist in **server.xml** file with the featurelist from the **server.xml** file in your analysis
      - The **server.xml** file in your analysis will be named: **APP_NAME.war_server.xml**
   10. You are now ready to follow the steps listed in the README.md to modernize to Containers
   11. Some files may contain the value APP_NAME. This must be replaced with your application name if the file is being used.

Frequently Asked Questions

Can I change Java versions while modernizing to Liberty in Containers? Yes you can. By default, we assume that the source and target Java levels are for Java 8. If for example you were using Java 8 and wanted to target Java 17 then you would use this command:

```
java -jar binaryAppScanner.jar <APP_NAME.war> --analyze --sourceAppServer=liberty
--sourceJava=ibm8 --targetJava=java17 --targetCloud=containers
```

Make sure that you always set your source Java correctly. For example if your existing application is using Java17 then your command will be:

```
java -jar binaryAppScanner.jar <APP_NAME.war> --analyze --sourceAppServer=liberty
--sourceJava=java17 --targetCloud=containers
```

There are other options available as well. Run the command with --help to see them.

Can I skip the analysis and just use the migration bundle? Yes you probably can. The analysis will be very useful if you have limited experience with containers, and want to become aware of the types of issues typically encounters. Many of the container specific issues will not stop your application from starting, but may lead to unexpected behaviour. For example, files created on disk by your application will be removed when the container restarts, unless they have been mounted on an external volume. This can lead to an intermittent and hard to find defects.

What dependencies does my application have? A limitation of this approach is that application dependencies will not be identified. You will need to manually add to the migration bundle any jar files that the application would normally get from shared library or classpath locations. The generic LibertyMigrationBundle does not provide the placeholder files for dependencies that a standard migration bundle from Transformation Advisor does.

My existing server.xml file has configuration for multiple applications, how do I find the features and configuration specific to this application? The features that this specific application needs will be listed in the server.xml file from your analysis. A limitation of this approach is that application specific configuration are not identified. This means that the original server.xml file may include configuration that this application does not need. However, the inclusion of unnecessary values will not stop your application from running.

Can I upload the results to Transformation Advisor? You can not. We are currently looking at how we can deliver some or all of this capability natively in Transformation Advisor

.

Modernization Target Comparisons

The table below compares all the modernization targets deployed as images, running in containers, across a number of different capabilities.

Capability	WebSphere Liberty	Open Liberty	WebSphere Traditional	Description
Spring Framework 4.x, Spring Boot 1.5	✓			Supported Spring and Spring Boot versions. Docs
Spring Framework 5.x, Spring Boot 2.x	✓	✓	✓	Supported Spring and Spring Boot versions. Docs
Spring Framework 6.x, Spring Boot 3.x	✓	✓		Supported Spring and Spring Boot versions. Docs
Java SE 8	✓	✓	✓	Supported
Java SE 11	✓	✓		Supported

Capability	WebSphere Liberty	Open Liberty	WebSphere Traditional	Description
Java SE 17+	✓	✓		Supported
Flexible Deployment	✓	✓	✓	Image based deployment allows for complete reproducibility and guaranteed immutability
Self-healing	✓	✓	✓	Container orchestration platform/operator restarts containers automatically
Horizontal scaling	✓	✓	✓	Container orchestration platform can scale deployments horizontally as needed, for suitable applications
High Availability	✓	✓	✓	Container orchestration platform can run multiple instance simultaneously to ensure availability
Zero Migration	✓	✓		Application code written for one version of the runtime is guaranteed to run on later versions of the runtime
Continuous Delivery releases	✓	✓		Continuously released every 4 week with latest security patches and features
Right size images	✓	✓		Runtime is optimised to include only required features therefore delivering smallest possible footprint
Fast Startup	✓	✓		Optimized for rapid startup in containerized environments
Auto-Tuning	✓	✓		Auto-tunes for smallest operational footprint
MicroProfile enabled	✓	✓		MicroProfile observability for health integration such as metrics, distributed tracing and readiness
Cloud-ready build	✓	✓		Build using Dockerfile, Cloud Native Buildpack, or Source2Image
Level 5 Operator	✓	✓		Full level 5 Operator for deployment and Day 2 Operations
Developer Mode	✓	✓		Enhanced developer experience optimised for Cloud
Latest Jakarta EE APIs	✓	✓		Supports the latest Jakarta EE APIs to keep applications as modern as possible

Capability	WebSphere Liberty	Open Liberty	WebSphere Traditional	Description
Supported Open Source		✓		Runtime source code is OpenSource. Full production level Support is available from IBM
Heritage WebSphere APIs	✓		✓	Support for existing WebSphere APIs to accelerate modernisation and reduce developer effort

Using the ACE data collector

ACE Fix pack 7 (11.0.0.7) provides a built-in Transformation Advisor command, which provides support for the analysis your existing IBM Integration Bus v10 integration nodes of any potential issues if you plan to move your architecture to adopt containers.

If you ran the TADataCollector command with the run parameter, a static HTML report is produced. The report lists any issues that are found for each integration server under the integration node.

The Overall Complexity Score is assessed as either Simple, Moderate, or Complex.

Since TA v 2.1.0, and ACE Fix Pack 10 (11.0.0.10), ACE data can be uploaded to TA. The archive also contains a HTML report that is produced by the ACE transformation advisor tool, see the ACE Knowledge Centre instructions:

https://www.ibm.com/docs/en/app-connect/12.0?topic=tasks-running-transformation-advisor-tool

Using the Transformation Advisor APIs

IBM Cloud Transformation Advisor provides a set of standard REST APIs.
The APIs are compliant with OpenAPI specification 3.0.

BETA release

The APIs provided are not yet fully complete and are subject to change. Transformation Advisor is not yet compatible with earlier versions in relation to APIs. There is no guarantee that future releases will support these APIs. The good news is that we can and will make changes for you. Just ask!

API documentation

The swagger documentation for the APIs is available from your Transformation Advisor installation at this location: **`<TA_SERVER>/openapi/`**
An interactive UI for these APIs can be found at: **`<TA_SERVER>/openapi/ui/`**

Finding the TA_SERVER value

The **<TA_SERVER>** value can be found from the OpenShift Cloud Platform using the following steps:

1. Launch the OpenShift Cloud Platform Console UI.
2. Click **Home > Projects** and select the project for Transformation Advisor.
3. Click **Networking > Routes** and select the `openapi-route`.
4. Find Location, which provides the value for `<TA_SERVER>`.

Limitations of Try it out

When running Transformation Advisor in OpenShift Cloud Platform with ingress enabled, the **Try it out** capability will fail on execution. This is because the generated curl command is missing the ingress value. You can add this value to the command and then run it manually to test the API.

Finding the TA_SERVER value with Transformation Advisor Local

The process for finding the **<TA_SERVER>** value is different if you are using Transformation Advisor Local (available from the IBM Garage Methodology website.

1. From the location where you run Transformation Advisor Local, issue this command: `docker ps`
2. Look for the image with `server` in the name and check the value of the port.
   - The default value is 2220, as in this example: `9443/tcp, 0.0.0.0:2220->9080/tcp`
3. Go to this URL: `<TA_LOCAL_UI_URL>:<SERVER_PORT>/openapi/ui/`

Server authentication

If authentication is enabled for Transformation Advisor, it is necessary to pass credentials with each API call. The swagger API provides the means to pass the necessary credentials.

Notes:

- In some cases the API will return a URL as a results.
- Usually it is not possible to use the URL directly as you will also have to provide credentials.
- In these cases you need to use the appropriate tools to add the credentials as headers.
- The exact format for the headers can be seen in the curl command the swagger API produces.

The required values to pass credentials are detailed in the following information:

API user token

To access the APIs using your own user token, complete the following steps:

1. Launch the OpenShift Cloud Platform Console UI.
2. Click on the user dropdown and select **Copy Login Command**.
3. Click **Display Token**.
4. Copy the value of the `token` attribute.
5. Enter this value into the apiKey field in the OpenAPI UI.

API key creation

To access the APIs without relying on a user token, complete the following steps to create a service account and use the API associated with that account.

1. Launch the OpenShift Cloud Platform Console UI.
2. Click on the user dropdown and select **Copy Login Command**.
3. Click **Display Token**.

4. Execute the Login command.
5. Create the service account, in this case called `ta-api`.

   ```
   oc create sa ta-api
   ```

6. Describe the service account.

   ```
   oc describe sa ta-api
   ```

7. This will list a set of Mountable secrets. Describe the first token secret.

   ```
   oc describe secret ta-api-secret-<TOKEN>
   ```

8. Copy the value of the `token` attribute.
9. Enter this value into the apiKey field in the OpenAPI UI.

API REST client

To easily integrate Transformation Advisor APIs into your product or tool, an open source REST client is being developed and will be available at a later date.

.

Data Migration

```
The recommended approach for any data migration is to upload the original zip
files that the Data Collector produced.
This can be done via the UI
If you no longer have those zip files, or you have a very large number of zip
files then use the steps outlined below
```

Note: Do not uninstall your existing version of Transformation Advisor until you have exported any data necessary for your migration.

The steps for data migration are slightly different depending on where you have deployed Transformation Advisor

Data Migration on Red Hat OpenShift Container Platform

Complete the following steps to migration your data

Back up your existing database

1. Export your data directory from the couchDB container
2. Export the data you currently have in Transformation Advisor
   - Use the bulk export API in TA 2.5. Full details can be found here
   - /advisor/v2/collectionArchives/bulkExport
   - /advisor/v2/workspaces/{workspaceId}/collectionArchives/bulkExport

Install Transformation Advisor

3. Install Transformation Advisor 3.0 on a different OCP cluster
   - If you do not have another OCP cluster available then complete the following steps

- Detach the `PersistentVolumeClaim` that is used by Transformation Advisor 2.5.x from the Transformation Advisor instance. Doing this will allow you to keep the Transformation Advisor data when you uninstall Transformation Advisor 2.5.x. If you need to revert to Transformation Advisor 2.5.x for any reason, you can re-install 2.5.x and specify the preserved `PersistentVolumeClaim` as an `existingClaim` in the configuration. Read Preserving the PersistentVolumeClaim for details on how to detach the `PersistentVolumeClaim`.
- Uninstall your existing Transformation Advisor 2.5.x
- Install Transformation Advisor 3.0

Import your data

4. Import your exported data into Transformation Advisor 3.0 using the bulkImport API. Full details can be found here
    - /advisor/v2/collectionArchives/bulkImport

Cleanup

5. If you have installed Transformation Advisor on a new cluster you should not delete your old Transformation Advsior instance

.

Backup/Export and Restore Transformation Advisor Data

NOTE: In your install configuration, if you have set `persistence.enabled` value to `false` then your data is stored only in the container and will be lost if the container restarts.

Persistent Volumes used by Transformation Advisor should be backed up in accordance with best practices.

Backup/Export of Transformation Advisor data

Option 1: Exporting using Transformation Advisor API

For Transformation Advisor on Red Hat OpenShift, using the Transformation Advisor API is the best option for backing up the data. See Exporting and importing Transformation Advisor data via HTTP endpoints document for more details on this option. In particular, see section entitled **Export all zip files for a workspace** that describes the bulk export API.

Option 2: Save data directories

Save data directories for Transformation Advisor Local

For a Transformation Advisor Local install, saving the data directories is the best option for backing up the data. You may also use the Transformation Advisor API to back up the data for a Transformation Advisor Local installation, but saving the data directory is likely to be more convenient. Go to the location where Transformation Advisor Local is installed.

Locate the **data** and **graph_data** directories and copy it to your desired backup location. You also need to copy the .neo4j_pass file.

```
cp -a <some location>/data <backup location>/data
cp -a <some location>/graph_data <backup location>/graph_data
cp -a <some location>/scripts/.neo4j_pass <backup location>/scripts/
```

Save data directories for Transformation Advisor on Red Hat OpenShift

It is not recommended to directly back up or restore data directories for Transformation Advisor on Red Hat OpenShift.

Restore/Import of Transformation Advisor data

Importing using Transformation Advisor API

If you have previously backed up your data using the Transformation Advisor API, you should use the API to import or restore that data to Transformation Advisor. See Exporting and importing Transformation Advisor data via HTTP endpoints document for more details on this option.

Restoring data directories

Restore data directories for Transformation Advisor Local

1. cd to the location of Transformation Advisor Local (i.e. the exploded Transformation Advisor downloaded zip)
2. Restore the data to the new location
   - `cp -a <backup location>/data .`
   - `cp -a <backup location>/graph_data .`
   - `cp -a <backup location>/scripts/.neo4j_pass scripts/`
3. In the new location install Transformation Advisor
   - `./launchTransformationAdvisor.sh`
   - Accept the license agreement
   - Choose Option 1 (Install)

Restore data directories for Transformation Advisor on Red Hat OpenShift

It is not recommended to directly back up or restore data directories for Transformation Advisor on Red Hat OpenShift.

Rolling back to a previous version of Transformation Advisor

When rolling back to a previous version of Transformation Advisor, it is important to use data that is compatible with that version. Transformation Advisor will not automatically convert new data to make it compatible with older versions of Transformation Advisor. For example, take the following scenario:

1. I install Transformation Advisor 3.0.0
2. I back up Transformation Advisor 3.0.0 data
3. I upgrade to Transformation Advisor 3.1.0
4. I back up Transformation Advisor 3.1.0 data
5. I want to roll back to Transformation Advisor 3.0.0

In this scenario, you must use the originally backed up data from the Transformation Advisor 3.0.0 installation. You cannot use the data backed up from the Transformation Advisor 3.1.0 installation.

Advanced Backup and Restore Options

The following advanced options should only be used if you are unsuccessful with other backup and restore options.

Preserving the PersistentVolumeClaim

You can perform an uninstall of Transformation Advisor but also preserve the **PersistentVolumeClaim** that it uses. This procedure is useful in the following scenarios:

- If you want to uninstall a version of Transformation Advisor and install a new version, but want an easy way to revert to the old version. This is recommended when moving from 2.5.x to 3.0.x.
- If you want to keep your data across Transformation Advisor version upgrades - when an automatic seamless upgrade is not possible (for example, when upgrading to 2.4.X form 2.3.X). Also, this approach will not allow you to preserve data when going from 2.5.x to 3.0.x.

The limitation with this approach is that you can only reuse the **PersistentVolumeClaim** in the *SAME* namespace it was originally used.

NOTE: This procedure should be used where the **PersistentVolumeClaim** that you want to preserve is one which was created automatically as part of a Transformation Advisor installation. If your Transformation Advisor install uses a **PersistentVolumeClaim** that was created manually, the following procedure might not be necessary in order to preserve the **PersistentVolumeClaim**. Run the following command on your **PersistentVolumeClaim**:

```
kubectl get pvc <my-pvc-name> -n <ta-namespace> -o yaml | grep ownerReferences
```

If that command shows no **ownerReferences** in the **PersistentVolumeClaim**, then you do not need this procedure, and you should be able to delete Transformation Advisor and your **PersistentVolumeClaim** will remain. Always backup data beforehand.

To preserve the **PersistentVolumeClaim**, follow the following procedure:

1. Remove the Transformation Advisor **subscription** and **clusterserviceversion**. This will also remove the operator. Replace **<ta-namespace>** in the following commands with the actual namespace where the install exists.

   ```
   export subscription=$(kubectl get subscription --no-headers -n <ta-namespace>
   | awk '{print $1}')
   kubectl delete subscription ${subscription} -n <ta-namespace> >/dev/null 2>&1

   export csv=$(kubectl get csv --no-headers -n <ta-namespace> | awk '{print
   $1}')
   kubectl delete csv ${csv} -n <ta-namespace>
   ```

Wait until the operator pod disappears before moving to step 2. Check the pods using **oc get pods**. Eventually the pod called **ta-operator-XXX** will disappear (where **XXX** is some random identifier)

2. Detach the **PersistentVolumeClaim** from the Transformation Advisor instance. This will allow us to delete the Transformation Advisor instance while keeping the **PersistentVolumeClaim**.

   ```
   export pvc_name=$(oc get deployment -n <ta-namespace> | grep couchdb | awk
   '{print$1}')
   kubectl patch pvc ${pvc_name} -n <ta-namespace> --type=json -p='[{"op":
   "remove", "path": "/metadata/ownerReferences"}]'
   ```

3. Remove the Transformation Advisor instance

```
# Run this command in the background using '&'
kubectl delete transadvs.charts.ta.cloud.ibm.com/ta -n <ta-namespace> &
kubectl patch transadvs.charts.ta.cloud.ibm.com/ta -p '{"metadata":
{"finalizers":[]}}' --type=merge -n <ta-namespace>
```

Wait until the Transformation Advisor pods have disappeared before going on the step 4. Check the pods using **`oc get pods`**. Eventually all the pods in the namespace will disappear.

4. Remove the **`CustomResourceDefinition`**

```
# Run this command in the background using '&'
kubectl delete crd/transadvs.charts.ta.cloud.ibm.com &
kubectl patch crd/transadvs.charts.ta.cloud.ibm.com -n <ta-namespace> -p
'{"metadata":{"finalizers":[]}}' --type=merge
```

5. Clean up other Transformation Advisor resources in the namespace

```
kubectl delete secret transformation-advisor-secret -n <ta-namespace>

export opertor_group=$(oc get operatorgroup -n <ta-namespace> --no-headers |
awk '{print $1}')
kubectl delete operatorgroup ${opertor_group} -n <ta-namespace>

kubectl delete clusterrolebinding <ta-namespace>-cluster-admin -n <ta-
namespace>
```

6. When you want to reuse the **`PersistentVolumeClaim`** that we have preserved, please read Configure storage and Installing for details on how to configure Transformation Advisor with an existing **`PersistentVolumeClaim`**.

   - If installing using the OpenShift UI, the persistence section of the custom resource YAML will look like something like the following. Make sure that the persistence properties are consistent with the **`PersistentVolumeClaim`** that you have preserved, for example the **`accessMode`** should be the same.

     ```
     ...
       persistence:
         enabled: true
         accessMode: "ReadWriteOnce"
         size: 8Gi
         useDynamicProvisioning: true
         existingClaim: "<my-ta-pvc>"
         storageClassName: ""
         supplementalGroups: []
     ...
     ```

   - If you are installing with the CASE installer, then you specify the existing claim using the **`--persistenceClaim <my-ta-pvc>`** argument.

7. **OPTIONAL:** After you have installed the new version of Transformation Advisor, you may attach the **`PersistentVolumeClaim`** to the new instance. Attaching the **`PersistentVolumeClaim`** to the new Transformation Advisor instance means that in future, if you want to uninstall the Transformation Advisor instance, the **`PersistentVolumeClaim`** will also be deleted automatically with the instance. This is also the default behavior of a standard Transformation Advisor installation. If you do not want this behavior, then you can skip this step. (Remember to substitute the actual namespace and **`PersistentVolumeClaim`** name into this command)

```
export uid=$(oc get transadvs.ta.ibm.com/ta -o yaml -n <ta-namespace> | grep
uid | awk 'NR==1{print$2}')

kubectl patch pvc <my-ta-pvc> --type=json -p='[{"op": "add", "path":
```

```
"/metadata/ownerReferences", "value": [{"apiVersion": "ta.ibm.com/v1",
"blockOwnerDeletion": true, "controller": true, "kind": "TransAdv", "name":
"ta", "uid": "'${uid}'"}]}]'
```

Recovery Point Objective

The most important Transformation Advisor data are the archives that are generated during a Transformation Advisor data collection. These archives should be backed up using a method of your choosing and stored for a time period appropriate for your organization.

The only data that cannot be restored from the original data collection archives is as follows:

- Manually created application groups
- Modifications to collection names

Once data has been loaded into the Transformation Advisor application, data can be retrieved and backed up using the API. The bulkExport API returns an archive of the Transformation Advisor data (*including modifications* that have been made to groups, workspaces, collections). That archive can be imported to a new installation using the bulkImport API.

Periodic backups using the bulkExport API are recommended. Alternatively, or in addition, you may also choose to periodically back up the database (see Back-up Options).

You should take the following factors into account when deciding the Recovery Point Objective:

- Frequency with which new data is uploaded to Transformation Advisor
- Frequency and volume of modifications to the uploaded data (creating, editing, deleting groups; editing workspace or collection names)
- Availability of backed up original Transformation Advisor data collection archives

.

Exporting and importing Transformation Advisor data via HTTP endpoints

This method does **NOT** involve locating and copying the underlying data from a CouchDB pod and hence can be helpful in these scenarios:

- Transformation Advisor is configured to use dynamically provisioned storage.
- Only some particular parts of the data need to be exported.
- The data to be exported is from an older version of Transformation Advisor running in IBM Cloud Private and it is to be imported to a newer version of Transformation Advisor running in OpenShift Container Platform.

How to export your data from Transformation Advisor in OpenShift Container Platform

First follow the guide Using the Transformation Advisor APIs to obtain the user token for authentication and the interactive UI for the OpenAPI endpoints.

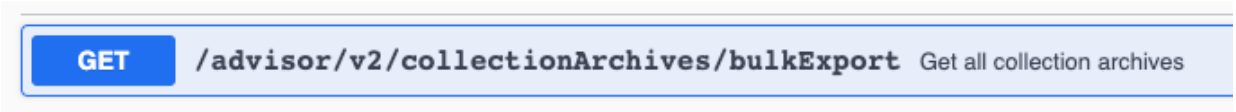
In addition, this method makes use of endpoints exposed on TA Web Server that are different from the OpenAPI ones. To locate the URL for these endpoints:

- Launch the OpenShift Container Platform Console UI.
- Click on Home > Projects and select the project for Transformation Advisor.
- Click on Networking > Routes and select the **`server-route`**.
- Find Location, which provides the value for **`<TA_WEB_SERVER>`** (e.g. at https://ta-ta.apps.apps.myinstance.host.com/lands_advisor)

Using the Swagger UI

Export all zip files

1. Use the **bulkExport** command in the Swagger UI to export all the zip files

2. If there is any issues with exporting all the zip files, follow the steps below to export the zip files for each workspace

Export all zip files for a workspace

1. Use the **workspace bulkExport** command in the Swagger UI to export all the zip files for the given workspace

2. Look up the **`workspaceId`** of the workspace via the **`GET /advisor/v2/workspaces`** OpenAPI endpoint and use this value

Using the Command Line

Export all zip files

To export the original zip files you can use the bulkExport API

1. Execute the following command: **`GET /advisor/v2/collectionArchives/bulkExport`**

Export all zip files for a workspace

To export the original zip files you can use the bulkExport API

1. Look up the **`workspaceId`** of the workspace via the **`GET /advisor/v2/workspaces`** OpenAPI endpoint
2. Execute the following command: **`GET /advisor/v2/workspaces/{workspaceId}/collectionArchives/bulkExport`**

This will return a zip file called: **`TA_bulkExport_<timestmap>.zip`**
The zip file has a hierarchical structure:

- At the first level is a directory for the workspace
- At the second level there is a directory for each collection
- Each collection directory will have 0-n zip files that were originally uploaded

How to export your data from Transformation Advisor in TA-LOCAL

For TA-LOCAL, the steps are similar to the previous steps except for the following:

- There is no need to obtain a user token for authentication.
- Remove the `--header="Authorization: Bearer <USER_TOKEN>"` part of all commands.
- The value for `<TA_WEB_SERVER>` is `<TA_LOCAL_UI_HOST>`:2220/lands_advisor

How to import your data to Transformation Advisor in OpenShift Container Platform or TA-LOCAL

Using the Swagger UI

First follow the guide Using the Transformation Advisor APIs to obtain the user token for authentication and the interactive UI for the OpenAPI endpoints.

Import more than 1 zip file at a time

You can import as many zip files as you like in one go by following the steps below

1. **Zip** all the zip files into a single zip file
2. Change the **name of this zip** file to be the name of the **workspace** you want to create
3. Use the **bulkImport** command in the Swagger UI to import all the zip files

   POST `/advisor/v2/collectionArchives/bulkImport` Upload bulk export generated from Transformation Advisor

4. Make sure you set the **noHierarchy** flag to be **true**
5. Attach the zip file and enter its name for the **bulkArchiveName**
6. Press **Execute**
7. As soon as the file is uploaded to the server the API will return with a **URL** (upload is complete but processing is still ongoing)
8. Open that **URL** to see the **progress** of the upload. **Refresh** this page to see further **progress**

Using the Command Line

Import a specific zip file

The simplest way is to upload the exported files via the UI of the target Transformation Advisor instance.

Alternatively, the original report archives and application dependency files can also be uploaded via an HTTP endpoint via the following steps:

- To upload the original report archives, go to the interactive UI of the OpenAPI endpoints and use the following REST method:

```
POST /lands_advisor/advisor/v2/workspaces/{workspaceId}/collectionArchives
```

Import all zip files for a workspace

If you have a bulkExport zip file you can use the bulkImport API to upload all the zip files at once.
The following command assumes you have renamed your bulkExport file to `bulkExport.zip` and it is located at `/root`

```
curl -X POST "
<TA_WEB_SERVER>/lands_advisor/advisor/v2/collectionArchives/bulkImport" -H
"accept: */*" -H "archiveName: bulkExport.zip" -H  "Authorization: Bearer
<USER_TOKEN>"  -H "Content-Type: application/octet-stream" --data-binary
"@/root/bulkExport.zip"
```

.

Disaster Recovery (DR)

NOTE: In your install configuration, if you have set **`persistence.enabled`** value to **`false`**, your data is stored only in the container and will be lost, if the container restarts.
NOTE: It's the best to back up your data regularly, see Backing up the CouchDB on how to do that, in particular the **`Recovery Point Objective`** section. Keep in mind, to recover your installation, you need to back up the config as well. **NOTE:** In this section we will assume your data is not lost completely and can be accessed by providing a path to it.

Backup

Data

Instructions on how to back up your data can be found here: see Backing up the CouchDB If you used **`existingClaim`** in your install, then PVs and PVCs need to be copied over into the new cluster before standing up new TA instance.
You can use code shown below to create yaml's for PV and PVCs. Apply them in the new cluster to create the objects.

```
oc -n <TA_NAME_SPACE> get pvc -o json | jq -Mr '.items[].spec.volumeName' | xargs
-I {} -n 1 oc get pv {} -o json | jq -Mrc 'del(.status,
.metadata.creationTimestamp, .metadata.uid, .metadata.resourceVersion,
.metadata.finalizers, .metadata.annotations, .spec.claimRef)' >> pv.yaml
```

```
oc -n <TA_NAME_SPACE> get pvc -o json | jq -Mrc '.items[] | del(.status,
.metadata.annotations, .metadata.creationTimestamp, .metadata.finalizers,
.metadata.ownerReferences, .metadata.resourceVersion, .metadata.uid)' >> pvc.yaml
```

TA config

Keep the original configs so that you can look them up while standing up new TA instance.

```
oc get transadvs.ta.ibm.com/ta -o yaml -n ta >> ta_instance_config.yaml
```

There is a good chance you won't be able to apply this yaml in the new cluster - some of the configs might change as they are cluster specific.
It's the best to create a new instance of TA in OCP UI (**`Create TransAdv`** page) and open the yaml view. Many of the configs will be populated by the pre-install process and you can adjust them using common sense and looking at the configs for the original TA instance.

Use existing data

OCP install

If running on OCP you can install a new TA instance pointing to the PVC you have created before. This is the config part where this can be set.

```
  persistence:
    couchdb:
      supplementalGroups: []
      accessMode: ReadWriteOnce
      size: 20Gi
      useDynamicProvisioning: true
      existingClaim: '<PVC_NAME>'
      storageClassName: ''
    neo4j:
      supplementalGroups: []
      accessMode: ReadWriteOnce
      size: 5Gi
      useDynamicProvisioning: true
      existingClaim: '<PVC_NAME>'
      storageClassName: ''
    enabled: true
```

Make sure you preserve the Access Mode and Size that you had before in case you have changed it.

TA Local

Provided the data remains on the machine, please follow instructions from Backing up the CouchDB in its "Restoring data directories" section.

Cluster configuration required

TA has no special requirements on the cluster configuration except storage (see Configure Storage).
Once TA config and data are backed up, you can use them in the new installation.
NOTE: There is a limitation on the egress network policy - it will not be enabled, if the OCP is not using OpenShift SDN default Container Network Interface (CNI) network provider.

Example config for a TA instance

TA provides two yaml's for **`Production`** and **`Quick Start`** configuration. They are available in the **`Create TransAdv`** page, **`YAML view`**, **`Samples`** tab on the right.

External services required to be available

None

.

Transformation Advisor API and CRD versioning

API versioning

To access OpenAPI URL with documented API for installed Transformation Advisor instance, go to https://openapi.<HOST_NAME>/openapi/ui/.

API Version	TA Version
V1	< 2.0.0
V2	2.0.0 or later

CRD versioning

To access CustomResourceDefinition in OCP where Transformation Advisor is installed, go to **Menu -> Administration -> CustomResourceDefinitions** and look for a CRD with a name **transadvs.ta.ibm.com**.

CRD Version	TA Version
V1	< 3.0.0
V2	3.0.0 or later

.

Licenses under which you have entitlement to use Transformation Advisor

Product name & version	License ID	Link to License
IBM WebSphere Hybrid Edition 5.1	L-ZZBG-6V3K4K	Review License
IBM Cloud Transformation Advisor 3.10.0 (Evaluation)	L-APJC-75QQ5K	Review License
IBM Cloud Pak for Integration 2023.2.1	L-YBXJ-ADJNSM	Review License
IBM Cloud Pak for Integration Limited Edition 2023.2.1	L-PYRA-849GYQ	Review License
IBM Cloud Pak for Applications Advanced 5.3	L-REBM-59QT8V	Review License
IBM Cloud Pak for Applications Standard 5.3	L-ZUVT-EDS78A	Review License
IBM Cloud Pak for Applications Limited 5.3	L-KULQ-TG6TN5	Review License
IBM WebSphere Application Server for z/OS 8.5.5	L-CTUR-C7K3YZ	Review License
IBM WebSphere Application Server for z/OS 9.0.5	L-CTUR-CBPUER	Review License
IBM WebSphere Automation v1.7.2	L-UMDR-KJVYBM	Review License

License Usage on Red Hat OpenShift Cluster

The following bash script can be used to retrieve license information for all the instances of Transformation Advisor on a cluster. The script can be run on RHEL or on MacOS. You must be `oc logged in` to the cluser as an admin to run the script.

```
#!/bin/bash

# ---------------------------------------------
# Licensed Materials - Property of IBM
# (C) Copyright IBM Corporation 2023
# ---------------------------------------------

#
# GLOBALS
#
report_name="./ibm-transformation-advisor-license-report.txt"

#
```

```
# FUNCTIONS
#

#
# Find the date 180 days in past
#
function get_180_days_ago() {
        local time_in_past_180=""

        # Check if the date command has the -j option
        if date -j >/dev/null 2>&1; then
                # Use the -j option for macOS
                time_in_past_180=$(date -j -v-180d +'%Y-%m-%d')
        else
                # Use the -d option for Linux
                time_in_past_180=$(date -d "-180 days" +%Y-%m-%d)
        fi

        # Print the Unix timestamp
        echo "${time_in_past_180}"
}


#
# MAIN ENTRY
#

time_in_past_180=$(get_180_days_ago)

# Check logged into OCP cluster
if oc whoami >/dev/null 2>&1; then
        current_user=$(oc whoami)
        current_ocp_cluster=$(oc whoami --show-server)
else
        echo "ERROR: You are not logged into an OpenShift cluster."
        echo "Please log into an OpenShift cluster as an admin user to run this
report"
        exit 1
fi

# Check if report file already exists
if [[ -f ${report_name} ]]; then
        read -r -p "A file called ${report_name} already exists. Overwrite?
[yes/no] " response
        case "$response" in
                [yY][eE][sS])
                echo "OK. Overwriting ${report_name}"
                rm -f ${report_name}
                ;;
                *)
                echo "Move, remove or rename ${report_name} before running this
tool."
                exit 1
                ;;
        esac
fi

echo "Generating report..."
echo -n "."


# Write report header
echo
"##############################################################################"
```

```
> $report_name
echo "# IBM Transformation Advisor License Report" >> $report_name
echo "#" >> $report_name
echo "# Time: $(date)" >> $report_name
echo "#" >> $report_name
echo
"##############################################################################"
>> $report_name

# Write report summary
echo "" >> $report_name
echo
"******************************************************************************"
>> $report_name
echo "SUMMARY" >> $report_name
echo "" >> $report_name
echo "User: $current_user" >> $report_name
echo "OpenShift endpoint: $current_ocp_cluster" >> $report_name

num_instances=$(oc get transadvs.ta.ibm.com --all-namespaces --no-headers | wc -l
| xargs)

if [[ ${num_instances} = 0 ]]; then
        echo "No instances of IBM Transformation Advisor were found on the
cluster. All namespaces were checked" >> $report_name
        exit 0
else
        echo "Number of IBM Transformation Advisor instances installed on the
cluster: ${num_instances}" >> $report_name
fi
echo -n "."

echo "" >> $report_name
echo
"******************************************************************************"
>> $report_name
echo "DETAILS" >> $report_name
echo "" >> $report_name

# Iterate over all the instance on the cluster
oc get transadvs.ta.ibm.com --all-namespaces -o custom-
columns=NAMESPACE:.metadata.namespace --no-headers | while read -r inst_namespace;
do

        echo -n "."

        num_instances=$((num_instances+1))

        namespace=${inst_namespace}
        name=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o custom-
columns=NAME:.metadata.name --no-headers)
        creation_timestamp=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o
custom-columns=CREATION:.metadata.creationTimestamp --no-headers)
        license=$(oc get transadvs.ta.ibm.com -n  ${inst_namespace} -o custom-
columns=LICENSE:.spec.license.aLicenseType --no-headers)

        echo "----------------------------------------" >> $report_name
        echo "Namespace: ${namespace}" >> $report_name
        echo "Instance Name: ${name}" >> $report_name
        echo "Created: ${creation_timestamp}" >> $report_name
        echo "License: ${license}" >> $report_name

        if [[ ${license} = *"Evaluation"* ]]; then
                creation_time=$(echo ${creation_timestamp} | sed "s/T.*//g")
```

```
                if [[ "${creation_time}" <  "${time_in_past_180}" ]]; then
                        echo "" >> $report_name
                        echo "WARNING: This instance was created more than 180
days ago and is using an evaluation license." >> $report_name
                        echo "Please upgrade to a full license." >> $report_name
                fi
        fi

        echo "" >> $report_name

done

echo "" >> $report_name
echo
"*******************************************************************************"
>> $report_name

echo ""
cat ${report_name}
echo "Report complete. Saved as ${report_name}"
echo ""
```

.

Data collector troubleshooting guide

Data collector and the Migration Toolkit for Application Binaries

The Transformation Advisor Data Collector uses a tool called the Migration Toolkit for Application Binaries. You may use the binary scanner directly. This may be particularly useful in some troubleshooting scenarios. For information on how to download and use the Migration Toolkit for Application Binaries please click here.

Failed to upload collection zip file to Transformation Advisor server

After the data collector gathers the configuration information for an application, it will create a zip file and try to upload it to the Transformation Advisor server. If it cannot reach the server, you may see the following error message:

```
Current Operation: Error occurred: Problem connecting with server See log for
details.
```

To work around this problem, you can get the collection zip file from the data collector directory and manually upload it to the Transformation Advisor server through the Collection page in the UI.

WebSphere profile <profile_name> does not exist

This message displays when the data collector cannot find the input profile name from the WAS profile registry. Verify that the input profile name is correct by checking the **WS_PROFILE_REGISTRY** defined in the `$WAS_HOME/properties/wasprofile.properties` file.

The default profile registry xml file is:
WS_PROFILE_REGISTRY=${was.install.root}/properties/profileRegistry.xml

If your profile registry xml file is not using the default file, create a symbolic link from **$WAS_HOME/properties/profileRegistry.xml** to the location of the real profile registry file.

Incorrect context root for web applications on WebSphere detected by data collector

In Transformation Advisor version 1.9.4 and later, the data collector detects the context root for web applications running on WebSphere. The context root of the application is used in deployments on Liberty to simplify access to the application. If the context root is not appropriate, a page with the following error message is displayed when accessing the deployed application:

```
Context Root Not Found.
```

To work around this problem, update the Helm chart for the application in the Git repository:

1. Clone the Git repository containing the Helm chart for the application.
2. Edit the file `chart/[application name]/values.yaml` and change the **rewriteTarget** to "/" as shown:

```
ingress:
  enabled: true
  rewriteTarget: "/"
```

3. Commit and push the changes to the Git repository.
4. Wait for the application to be re-deployed. The ingress path for the application now redirects to the Liberty base page. You can access the application by appending the context root of the application to the ingress path.

For example, if the ingress path is *modresorts* and the context root for the application is *resorts,* the URL to access the application would be **http://[OCP public IP]/modresorts/resorts/**.

Data collector fails to generate server.xml artifacts

If the data collector is run by a user other than the user that has launched the deployment manager, it will fail to generate the server xml artifacts. The collected artifacts will look like a collection that has not requested configuration analysis. On Transformation Advisor versions earlier than 1.9.4, you may see the following error:

```
Exception in thread "main" java.io.FileNotFoundException: <some_path>_server.xml
(A file or directory in the path name does not exist.)         at
java.io.FileInputStream.<init>(FileInputStream.java:113)         at
java.io.FileInputStream.<init>(FileInputStream.java:73)                 ...
...
```

You must run the data collector as the same user who has launched the deployment manager. You can check which user that is by using the **ps** command.

Note: If you have already unpacked the data collector as a different user, you may need to edit the permissions or owner for the unpacked data collector directory before you run it as the new user. Alternatively you can simply unpack the data collector again as the new user.

Checksum error trying to extract the data collector

If a checksum error is returned after trying to extract the data collector, try this workaround:

```
gzip -d transformationadvisor-<OperatingSystem>_<workspace>_<collection>.tgz
tar xf transformationadvisor-<OperatingSystem>_<workspace>_<collection>.tar
```

Error when running the data collector in non-English locales

If you see **`Connecté au processus dmgr sur le noeud ...`** (or something similiar) when running the the data collector with the command **`/data/WebSphere/AppServer/profiles/Dmgr01/bin/wsadmin.sh -user wasadmin -password wasdm1i -lang jython -c "AdminApp.list()"`**, try the following work around:

In the terminal where you are running the data collector, type:

```
export LANG=C
```

Then run the data collector command again and it should bypass the problem.

The binary scanner ignores com.ibm packages

The data collector can be configured to ignore certain packages. Check the **`customCmd.properties`** file in **`$TA_HOME/conf`** directory to verify. If there are any packages that are ignored, you can uncomment the line and modify the command to include them. The data collector will include uncommented commands when invoking the binary scanner.

The data collector skips managed profiles when run against a specified profile

Managed profiles are skipped because a copy of their configuration is held on the dmgr. Running on the dmgr should solve this issue.

The data collector skips the applications that were installed using zero binary copy mode when run against a specified profile

Applications that were installed with zero binary copy mode are skipped because the data collector can't get all of the required configurations for the applications to generate the migration analysis. This type of application must be manually evaluated to determine the migration effort.

Running the data collector produces a `libjvm.so failed to load:` error

This error can be caused when Java cannot find the correct libraries. Use the Java version on the machine itself instead of the one that is downloaded with the data collector. Complete the following steps to resolve this error:

1. Check to see if the **wsadmin** is owned or run by a specific user. If it is, then that same user should run the data collector.

2. Make sure that the user has read, write, and execute permissions to the location where you have unpackaged the data collector.

Make sure you are using the version of Java that is on the machine:

1. cd into transformationadvisor-2.1
2. Replace the JRE directory with the JRE directory from the WAS machine itself.
3. Run the data collector command again.

Error during upload - Incompatible files detected

The data collector produces reports that are linked to each application. If you see this error, it means that one or more applications were not processed correctly and the step to link the report to an application has failed.

You can resolve this issue as follows:

1. Unpackage the `profile.zip` file.
2. Delete files with the following names: `InventoryReport.json, InventoryReport.html, AnalysisReport.json, AnalysisReport.html, TechnologyReport.json, TechnologyReport.html`
3. Manually upload the zip file to Transformation Advisor through the UI.

To determine if you are missing any applications:

1. Get the list of applications in Transformation Advisor.
2. Look at the profiles directory in WAS_HOME to review the full list of applications installed in that profile.
3. Identify applications in the profile that are missing from Transformation Advisor.

Untar failed after downloading the data collector and copying it over to a Linux VM

Try the following commands instead:

```
gzip -d transformationadvisor-2.1_Linux_xxxxxx.tgz
tar xf transformationadvisor-2.1_Linux_xxxxx.tar
```

Unable to download the migration files after data upload

If you upload a zip file from the data collector to the Transformation Advisor UI but are unable to download the migration bundle or any migration files, you'll need to run a full analysis so that Transformation Advisor can get the server configuration. Run the following command:

```
bin\transformationadvisor.bat -w <WEBSPHERE_HOME_DIR> -p <PROFILE_NAME>
```

Then try the upload again.

Windows data collector does not work if WebSphere home contains a space in the path

If you run the Windows data collector with the following command:

```
bin\transformationadvisor.bat -w "C:\Program Files (x86)\IBM\WebSphere\AppServer"
-p AppSrv01
```

The data collector might throw an error message. Complete the following to resolve the issue:
Open command prompt and go to the directory which you want to know the short path

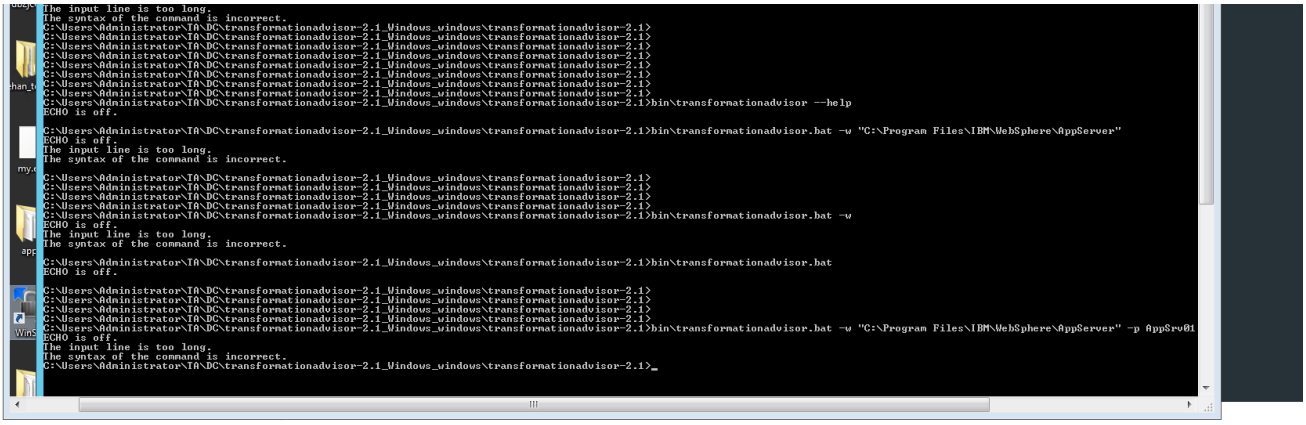
Type dir /x

Run the data collector with the shortcut name, for example:

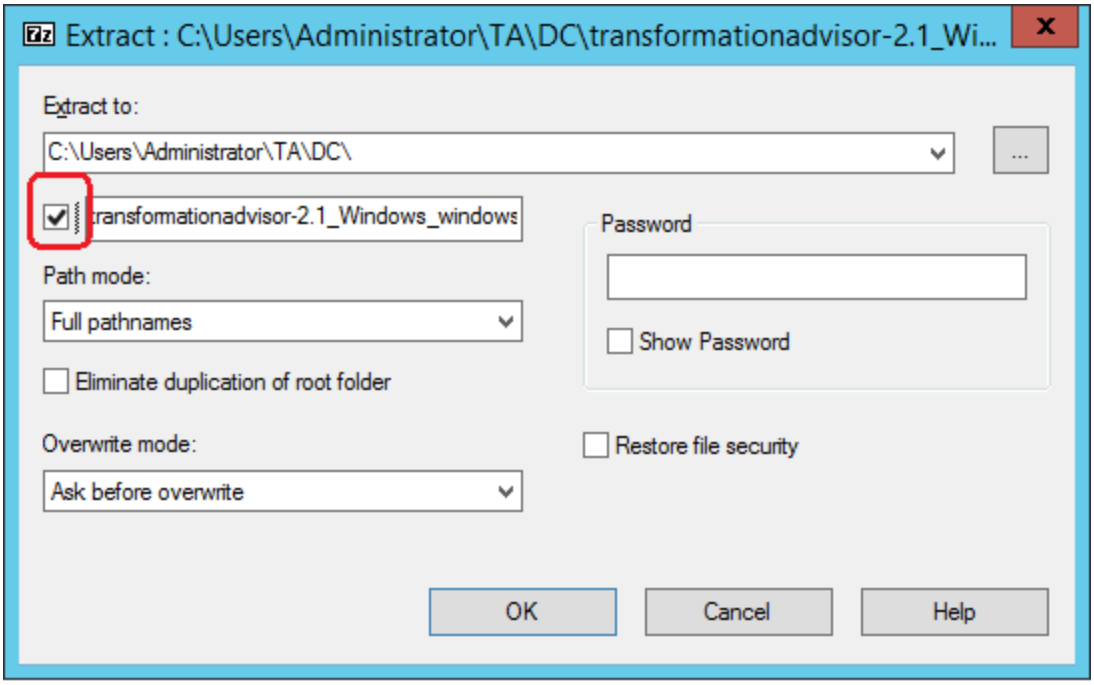
```
bin\transformationadvisor.bat -w "bin\transformationadvisor.bat -w
"C:\PROGRA~2\IBM\WebSphere\AppServer" -p AppSrv01
```

Windows data collector does not work after extracting to default path

If you run the Windows data collector with the **`bin\transformationadvisor.bat -w "C:\Program Files\IBM\WebSphere\AppServer" -p AppSrv01 admin admin`** command and see the following output:



The path for TA_HOME is too long. To work around this issue, uncheck the option to use the file name as the subdirectory as the path already contains the transformationadvisor-2.1 subdirectory when you unpackage the data collector zip file. When this box is unchecked, the data collector will unzip to:
`C:\Users\Administrator\TA\DC\transformationadvisor-2.1`



Steps to run the data collector on a z/OS environment

NOTE: Running the data collector on a z/OS environment has been tested with bash versions 4.2 and 4.3

1. It is recommended that the data collector be run from an ssh session, and not the Unix System Services OMVS shell since the data collector uses screen refresh in ways that don't work on 3270-type devices.
2. Install bash if it's not already available and make sure it is added to the system path, e.g. `export PATH=/usr/bin/rocket/bash-4.3/bin:$PATH`
3. Install gunzip if it's not already available, e.g. from Rocket Software
4. sftp / download the data collector file `transformationadvisor-zOS.tgz` to the z/OS machine
5. Extract the `.tgz` file, e.g. by running: `gunzip -c transformationadvisor-zOS.tgz | tar xf -`
6. Ensure that the ownership of the transformationadvisor-2.X.X folder is that of the user that will later run the data collector. If necessary, change permissions and ownership of the transformationadvisor-2.X.X folder by running:

```
chown -R <user> transformationadvisor-2.X.X
```

7. Change directory to the transformationadvisor-2.X.X folder, add that to the system path
8. Run (from regular shell, NOT from bash): `cd bin && . ./zOSPrereq && cd ..`
9. To see all the run options: `bin/transformationadvisor --help`. The first time the data collector runs, there will be a license agreement text to review. If a memory-related issue occurs, add the `-J-Xmx512m` option, i.e. `bin/transformationadvisor -J-Xmx512m --help`

10. (This step is not required from TA 3.0 onwards) If user defined rules are used (a feature only available from TA 2.5.0), note that all files within the `conf/userDefinedRules` are in UTF-8 and need to be kept in that encoding. Therefore, to use this feature in z/OS, work with the files in UTF-8 in another system and then copy them back in.
11. Run the data collector, for example: `bin/transformationadvisor -w <WAS home>` or `bin/transformationadvisor -w <WAS home> -p default`
12. The data collector will produce a zip file, e.g. default.zip, that can be uploaded to the IBM Transformation Advisor UI for viewing the results.

.

Running TA Data Collector from a copy of WebSphere config

In order to run the Transformation Advisor data collector against a backup copy of the configuration, the specific profile folder is needed as well as the properties folder.

To view the profile location path, view the contents of the file below

```
<WAS_HOME>/properties/profileRegistry.xml
```

In this example, the copy is in the BACKUPWAS folder.

```
./BACKUPWAS/
 ├── profiles
 └── properties
```

zip the profiles/Dmgr01 folder and copy that to BACKUPWAS/profiles/Dmgr01 zip the properties folder and copy that to BACKUPWAS/properties

for each profile used in the backup, update the location of the profile in the file

```
BACKUPWAS/properties/profileRegistry.xml
```

run the data collector with the following command:

```
./bin/transformationadvisor -w /opt/IBM/BACKUPWAS/ -p Dmgr01
```

Shared Libraries

If the profile uses Shared Libraries that cannot be accessed from where you are running the scan on the configuration BACKUP, you will need to copy the shared libraries to the backup folder and update the location of each shared library in the BACKUPWAS folder's config.

To view the shared library location paths:

- Login to the WebSphere Admin console
- Expand Environment
- Click the Shared libraries link to see the location Note: If the Classpath is defined as a value you can get the location by clicking the WebSphere variables link and searching for the name of the classpath
- For each shared library used, update the location of the SL by either: updating BACKUPWAS/profile//config/cells//libraries.xml when the library location is hard coded OR BACKUPWAS/profile//config/cells//variables.xml when the library location uses a variable.

- Run the data collector with the following command:

```
./bin/transformationadvisor -w /opt/IBM/BACKUPWAS/ -p Dmgr01
```

.

Security Troubleshooting guide

NOTE: Please refer to Installation document to understand how to apply configurations for your given install, it will be different depending on your install option either installing from the OpenShift UI, or using the CASE installer. If installing as part of Cloud Pak for Applications, use the **`data/transadv.yaml`** file to perform the configuration.

Environment variables missing

In order to ensure Transformation Advisor security, certain environment variables are required. These variables are set automatically when Transformation Advisor is installed. If any of the values are missing, you will see this message from the UI server logs:

```
Authentication is set to enable, but some required environment variable(s) is
missing. This may cause Readiness probe failure. Please uninstall TA, re-configure
TA and try again
```

One or more of the following variables may not be set correctly:

```
TA_OCP_API_ENDPOINT=https://example.eu-de.containers.cloud.ibm.com:30999
TA_AUTH_ISSUER_ENDPOINT=https://example.eu-de.containers.cloud.ibm.com:30999
TA_AUTH_OIDC_CLIENT_ID=example5282946fac07867fbc937548cb35d3ebbace00
TA_AUTH_OIDC_CLIENT_SECRET=example5282946fac07867fbc937548cb35d3ebbace00
TA_API_KEY=B513VT1zlF56J9ZBjdambQQVM5_jePVN
TA_AES_IV=n6Ma2S4DiYVK3AlM
TA_AES_KEY=B513VT1zlF56J9ZBjdambQQVM5_jePVN
TA_DB_USER=admin
TA_AUTH_OIDC_CALLBACK_URI=http://example-ui-server/auth/callback
# the value of the private key shall be the base64 encoded pem format key
TA_PRIVATE_KEY
# the value of the public key shall be the base64 encoded cert
TA_PUBLIC_KEY
```

You can find this list of variables by running the following command:

```
oc exec -ti <server_pod_name> env | grep TA
```

To manually configure the environment variables, you need to update the **`data/transadv.yaml`** file that is provided by the installer:

1. Uninstall Transformation Advisor.
2. Re-configure data/transadv.yaml.
3. Install Transformation Advisor.

For example, to create a test public key and private key:

```
# Run this only once and make sure that you save and don't lose the files:
openssl req -newkey rsa:2048 -new -nodes -keyout private.pem -out unsigned-
public.pem -subj "/C=IE/ST=Cork/L=Cork/O=IBM/CN=www.ibm.com"
openssl x509 -req -days 730 -in unsigned-public.pem -signkey private.pem -out
```

```
public.crt

# An example to base64 encode the key pairs, depending on the base64 version on
your VM:

# Red Hat from 2018
# Do `man base64` and last line: Redhat: GNU coreutils 8.22 June 2018: Wrap
encoded lines after COLS character (default 76). Use 0 to disable line wrapping.
base64 -w 0 ./private.pem > private-base64
base64 -w 0 ./public.crt > public-base64
```

The values of private-base64 and public-base64 are the values of **TA_PRIVATE_KEY** and **TA_PUBLIC_KEY.**

Some values are generated by the installer and are not configurable in the **data/transadv.yaml** file.

Authentication page inaccessible

If you reconfigure the **data/transadv.yaml** file while troubleshooting and then cannot get to the authentication page, check that the OAuth callback URL is set correctly.

In the following example, the client ID is:
TA_AUTH_OIDC_CLIENT_ID=example5282946fac07867fbc937548cb35d3ebbace00

```
    ocp:
      # TA_AUTH_ISSUER_ENDPOINT
      authIssuerEndpoint: "{{ cluster.authorizationEndpoint |
regex_replace('/oauth/.*', '') }}"
      # TA_OCP_API_ENDPOINT
      apiEndpoint: "{{ clusterUrl }}"
      # contains the data that configure the: TA_DB_USER and TA_API_KEY
      secretName: "transformation-advisor-secret"
    oidc:
      endpointPort: "{{ cluster.authorizationEndpoint | urlsplit('port') }}"
      # TA_AUTH_OIDC_CLIENT_ID
      clientId: ta-ui
      # TA_AUTH_OIDC_CLIENT_SECRET
      clientSecret: "{{ lookup('password', '/dev/null length=40 chars=hexdigits')
}}"
```

Ensure that you have the Red Hat OpenShift Container Platform connected to your terminal or can initialize the environment in another way and issue the following command:

```
oc edit oauthclient example5282946fac07867fbc937548cb35d3ebbace00
```

If the redirect URL does not exist or is wrongly configured, update it to the correct URL. For example:

```
redirectURLs:
- http://your-ta-ui-url/auth/callback
```

Disabling authentications

You can disable authentication for debugging or testing purposes by re-configuring the **data/transadv.yaml** file:

1. Uninstall Transformation Advisor.
2. Re-configure **data/transadv.yaml** to update authentication:

```
  authentication:
    disabled:
```

```
      liberty: true
      ui: true
```

3. Install Transformation Advisor.

Key rotation

Red Hat Provided Certificates and Keys

The certificates (referred as certs in the rest of section) and keys are stored in the secrets:

```
# used by the couch db pod
db-internal-cert

# used by the Liberty server pod
liberty-internal-cert

# used by the UI pod
ui-internal-cert
```

To replace the cert and key pairs used by internal TLS by Transformation Advisor (TA), complete the following steps:

1. Delete the secret used by the pod.

   Always backup the data, before removing the Couch DB pod, if the Transformation Advisor is installed without a Persistent Volume (PV).

2. Delete the pod.

For example, to replace UI pod's cert and key,

```
# delete the secret
oc delete secret ui-internal-cert

# delete the pod
oc delete pod ui-pod-name -n ta

# always delete the liberty pod
oc delete pod liberty-pod-name -n ta
```

To get the pod's name:

```
oc get pods -n ta
```

Verify The Key rotation

You can verify the change of the cert by logging in to the liberty pod and connect to the UI pod:

```
oc exec -ti liberty-pod-name bash

# in the liberty pod
openssl s_client -connect ui-service-name:3443
```

Then, compare the `Server certificate` section before and after the key rotation.

To get the service's name:

```
oc get services -n ta
```

Customers Provided Certificates and Keys

Follow the steps to replace customers provided certs and keys

1. Backup data
2. Uninstall Transformation Advisor
3. Delete TA secret `transformation-advisor-secret`, if any.
4. Follow the Enable Bring Your Own Key (BYOK) section and use a new cert and key pair.

Reset Internal Credentials

All internal credentials are stored in the secret `transformation-advisor-secret`.

Follow the steps to reset internal passwords, key, initial vector etc:

1. Back up your old secret: `oc get secret transformation-advisor-secret -o yaml > backup.yaml`

2. Delete the secret: `oc delete secret transformation-advisor-secret`

3. Re-create the credentials. Follow the Enable Bring Your Own Key (BYOK) section to create a new secret. For example,

```
 oc create secret generic transformation-advisor-secret \
 --from-literal=db_username='updated-plain-text-username' --from-
literal=secret='updated--text-password' \
 --from-file=ta_public_key=./public-base64 --from-
file=ta_private_key=./private-base64 \
 --from-literal=ta_aes_key=$TA_TEMP_KEY --from-literal=ta_aes_iv=$TA_TEMP_IV
\
 --from-file=key.p12=key.p12
```

4. Delete the CouchDB pod, UI pod and then the Liberty pod.

```
 # delete the pod
 oc delete pod couchdb-pod-name -n ta

 oc delete pod ui-pod-name -n ta

 oc delete pod liberty-pod-name -n ta
```

Invalid Certificate Issue on OCP on IBM Cloud (ROKS)

If you already have a valid certificate installed on IBM Cloud, and the certificate is also valid on OCP web console page on IBM Cloud.

You can enable this certificate to be valid for Transformation Advisor by changing Transformation Advisor's routes from `ta.apps` to `ta-apps` before you install Transformation Advisor.

You can change the routes with the following steps:

1. In the `Installed Operators`, click IBM Transformation Advisor.
2. In the `Installed Operators > Operator details` page, click `Create instance`.
3. In the `IBM Transformation Advisor > Create TransAdv` page, go to `YAML view`.
4. In the `YAML view` editor, search for `ta.apps`, and change to `ta-apps`.
5. Click **Create** in the same page.

Egress Network Policy (ENP)

A TA provided Egress network policy is enabled by default as of Transformation Advisor 3.1.0. in keeping with Kubernetes security hardening principals to limit pod external communication. If the OpenShift SDB plugin is not installed, the TA egress network policy will be disabled,.

The TA default ENP denies outbound communication from Transformation Advisor's installation namespace, with the exception of DNS names github.com and github.ibm.com.

If desired, the TA provided Egress network policy can be disabled. Disabling the TA provided network policy allows TA pods non-isolated (or unrestricted) outbound communication

You can modify the ENP before or after the Transformation Advisor installation.

Modify Egress Network Policy before Installation

1. Install Transformation Advisor Operator.
2. Click *Create Instance* inside IBM Transformation Advisor operator detail page
3. Go to *YAML view*, you should see something similar to the following code:

```
networkPolicy:
  enabled: true
  egress:
    enabled: true
    default:
      - type: Allow
        to:
          dnsName: github.com
      - type: Allow
        to:
          dnsName: github.ibm.com
      - type: Allow
        to:
          cidrSelector: 10.254.0.0/16
      - type: Allow
        to:
          cidrSelector: 10.17.76.179/32
      - type: Allow
        to:
          dnsName: api.fuguo.ken.fyre.ibm.com
```

4. To disable ENP, change `networkPolicy.egress.enabled` to `false`.
5. To allow or deny DNS name, add similar entries:

```
- type: Deny
  to:
    dnsName: this.dns.name.will.be.denied
- type: Allow
  to:
     dnsName: this.dns.name.will.be.allowed
```

6. To allow or deny CIDR ranges, use `cidrSelector` keyword with CIDR value to replace `dnsName` and its value.

Modify Egress Network Policy after Installation

1. Navigate through *Edit TransAdv* and go to *TransAdv details* page in *Installed Operators > Operator details*.
2. Click the *YAML* tab.

3. Follow the same instruction to modify the ENP in the **Modify Egress Network Policy before Installation** section.
4. The ENP will take effect after the Transformation Advisor pods are re-created by the operator.
5. If the pods' status does not change after a while, you can manually delete the Transformation Advisor pods.
6. To check the ENP, you can use the command `oc describe egressnetworkpolicy ta-default-egress-network-policy -n your-ta-name-space`

.

Frequently Asked Questions when using IBM Cloud Transformation Advisor

Data collector

How do you exclude large files to improve the data collector execution time?

1. Find the `customCmd.properties` in the `./transformationadvisor-{version}/conf` directory.
2. Update the `customCmd.properties` file with the files you wish to exclude. For example: `--excludeFiles='.*/largeFile.xml'`
3. Run the data collector.
   Note: The log files will show the files that are being excluded.

How do you enable verbose logging for the binary scanner?

You can enable verbose logging for the binary scanner tool that the data collector runs. Verbose logging is available in trace files which are generated in the directory where you run the data collector. To enable verbose logging, use the `--verbose` option on the command line.

For example:

```
./bin/transformationadvisor -w /opt/IBM/WebSphere/AppServer/ -p AppSrv01 admin --verbose
```

When I have multiple versions of the same .EAR deployed, why do I only see one in the analysis table?

If there are multiple versions of the same application on your WebSphere server the data collector will only scan and provide analysis results on one of those versions.

How do I define a customer SOAP Timeout for an execution?

1. Copy the `soap.client.props` file from `<PROFILE_ROOT>/properties` into the data collector configuration directory.
2. IMPORTANT: Update the name of the `soap.client.props` file to: `tasoap.client.props`
3. Edit `tasoap.client.props` and update the value of `com.ibm.SOAP.requestTimeout` as required.
4. Rerun the data collector.

How do I increase the system default for SOAP timeout?

1. Find the `<PROFILE_ROOT>/properties/soap.client.props` file.

2. Increase the value for the property `com.ibm.SOAP.requestTimeout`
   Note: If you set the value to 0 there will be no time limit.
3. Rerun the data collector.

How can I set the heap size in the data collector when there is an Out of Memory issue?

You can use the `-J-Xmx` option in the command line to customize the maximum heap size used by data collector. For example, to set the maximum heap size to 2 GB:

```
./bin/transformationadvisor -J-Xmx2G -w /opt/IBM/WebSphere/AppServer/ -p AppSrv01
admin
```

How do I scan a Custom Profile?

Transformation Advisor will automatically find your custom profiles if you supply the correct profile name. If for some reason this is not working, you can create a symbolic link from the profiles directory in WAS_HOME to the location of the custom directory using this command:

```
ln -s <CUSTOM_PROFILE_LOCATON>/<CUSTOM_PROFILE_NAME>
<WAS_HOME>/profiles/<CUSTOM_PROFILE_NAME>
```

For example:
A custom profile called appProfile has been created outside of WAS_HOME:

```
ln -s /was7/appProfile/ /ibm/was/profiles/appProfile
```

Can you run the data collector with a specific JRE?

By default, the data collector automatically determines which version of Java to use based on the following rules:

1. If collecting from a WebSphere Application Server, the version of Java that is installed with the WebSphere Application Server will be used.

If you are not collecting from a WebSphere Application Server:

2. The version of Java that is on the path will be used.
3. If there is no version on the path, then the version of Java that is shipped with the data collector will be used.

If you want to use a specific JRE for the data collector, you can include a `--java-home option` in the command. This option will override the defaults outlined previously. For example:

```
./bin/transformationadvisor --java-home /opt/ibm/java-x86_64-80
-w /opt/IBM/WebSphere/AppServer/
-p AppSrv01 admin
```

Does Transformation Advisor use the version of Java provided by the WebSphere Application Server installation?

The data collector will attempt to use the Java version that is provided by the WebSphere Application Server. If for some reason it is unable to do so then it will use the version that is packaged with it. In any situation where that will not work, you can copy in your local JRE by replacing the JRE directory under transformationadvisor-2.1 with the JRE directory that the WebSphere Application Server is currently using, which is found in the WebSphere Application Server machine.

What exactly are ‘Collections’ and how should I name them?

You can name collections anything that you find helpful to organize work under your workspace.

For example, a WebSphere Application Server can have many profiles. Each of these profiles may have many applications. So when you run the data collector against a WebSphere Application Server, you will potentially get multiple applications for each profile, from just one server.

You may also have more than one server in total. So you could name the collection the same name as the WebSphere Application Server that you are scanning and under that collection you will then have the multiple profiles and applications for the server on which you ran the data collector.

How can I scan my profile if it is not in WAS_Home?

If for example the profile name is **`AppSrv010`** and it is under **`/root/was_profile`**, you can create a symbolic link to the profile as follows:

```
ln -s /root/was_profile /opt/IBM/WebSphere/AppServer/profiles/AppSrv010
```

What can I do if the server restarts while data is uploading on TA LOCAL?

If the server restarts while data is uploading it could indicate the zip file is large and the server needs more memory:

Increase the memory resource in Docker Desktop from 8GB to 12GB

How do I use customized commands for the binary scanner?

Customized commands for the binary scanner can be issued by using the **`customCmd.properties`** file in the conf directory. There is a different command for each report.

The value of **`java_opt`** is used to customize the JRE runtime for every binary scanner command. By default, all the commands are commented out. You can uncomment the commands which you want to customize.

The data collector uses the complete command when generating that specific report. If any of the custom command options are incorrect, the data collector will stop executing and throw an error.

Guidelines for writing the custom commmand options

1. Don't specify the **`--format option`**. The Transformation Advisor server expects both html and json formats for each report. Generating only one report format may introduce errors while uploading and presenting the results.
2. Don't specify the **`--output option`**. The Transformation Advisor server expects file names in a certain order. Also, specifying this option may change the location of reports storage which may cause problems while zipping up and uploading the results.
3. If an option needs a path as its value then always surround the paths with double quotes: **`path="c:\\docs\\doc1"`**
4. The backslash character must be escaped as a double backslash: **`path="c:\\docs\\doc1"`**

How do I collect data for a large set of applications on WAS?

If there are more than 100 applications deployed on multiple WebSphere Application Server profiles, you can invoke multiple data collector processes, one for each profile.

If there are more than 100 applications deployed on the same WAS profile, you can use the **--applications** option which will take a list of application names or the **--applicationsFile** option which will read the application names from a file.

Can I upload data collected using the Migration Toolkit for Application Binaries?

Yes, an alternative to using the Transformation Advisor Data Collector is to us the the Migration Toolkit for Application Binaries to generate a data collection that can be uploaded to IBM Cloud Transformation Advisor. Details of how to do this are outlined here.

Where can I find out more about the binary scanner and the server configuration files generated?

You can find out more about the binary scanner at https://www.ibm.com/docs/wamt. For more information on the binary scanner generated configuration files see https://www.ibm.com/docs/wamt?topic=binaries-configuration-migration

General

What versions of WebSphere Application Server can Transformation Advisor support?

Version 7+. On version 6.1, the data collector can scan the binaries and provide an analysis, but cannot get the configuration and create a migration bundle for you.

Should I use Nodeports to expose my endpoints in a production configuration?

Nodeports are a simple way of exposing external access to a workload for initial development and testing but they expose additional security concerns and are hard to manage from both an application and networking infrastructure perspective. We would recommend you use Routes when deploying on Red Hat OpenShift Container Platform and Ingress on IBM Cloud Private.

What does support for JBoss and WebLogic mean?

The Transformation Advisor data collector scans for known issues in applications that are migrated from WebLogic Servers and JBoss Application Servers to WebSphere Application Servers. Where possible, a quick fix is suggested to change your code to a more portable solution.

You can use the quick fix to help you decide if you want to accept the suggested code change. Also, view the help information provided with the complexity rules to decide if you want to run the quick fix. Always make a backup copy of your source code before you start a migration.

For some rules, the scan detects code that requires design changes and code rewrites. The tools highlight these problem areas but do not provide a quick fix.

What versions of WebLogic can Transformation Advisor support?

Transformation Advisor supports runtime migrations from these Java EE servers:

- WebLogic Server 6.x – 11.x to Liberty or Full profile
- WebSphere Application Server V7.0 (and later) to Liberty or Full profile

Is it possible to upload the Transformation Advisor results zip file using a REST call?

Yes, you can use the `curl` command:

```
curl -X 'POST' -H 'accept: /' -H 'archiveName: <zip file>' -H 'locale: en' -H
'Content-Type: application/octet-stream'      'http://<host>:
<port>/lands_advisor/advisor/v2/workspaces/{workspaceId}/collectionArchives?
overwrite=true'      --data-binary "@/<path-to-file>/<zip file>"
```

Does Transformation Advisor provide anti-virus scan?

No. Please ensure all the uploading files are trusted, or pre-scanned by an anti-virus software.

Why Transformation Advisor UI's logout does not work after I upgrade to OpenShift Container Platform 4.6?

It is a known issue introduced by OpenShift Container Platform 4.6. More info at [Secure OAuth token storage format](#).

TA users on OpenShift Container Platform can revoke the access tokens manually by following steps in the terminal:

1. Assumed you have permissions to access OCP via terminal with **`oc login`** command and the following commands,

2. Get Transformation Advisor's **`oauthclient`** name:

```
> oc get oauthclient -l=app.kubernetes.io/name=ta-oauth
```

```
NAME                                          SECRET
WWW-CHALLENGE   TOKEN-MAX-AGE   REDIRECT URIS
ca5282946fac07867fbc937548cb35d3ebbace7e
94b6cbce793d0606c0df9e8d656a159f0c06631b   false           default
https://ta.apps.example.ibm.com/auth/callback
```

   `ta-oauth` is in the format of **`{ .Release.Name }-oauth`**. Please update the release name accordingly, if necessary.

   The output name **`ca5282946fac07867fbc937548cb35d3ebbace7e`** is the Transformation Advisor **`oauthclient`** name, which will be used in the next command.

3. Get a list of access tokens under the **`oauthclient`** name:

```
# change the name ca5282946fac07867fbc937548cb35d3ebbace7e accordingly
```

```
> oc get oauthaccesstokens | grep ca5282946fac07867fbc937548cb35d3ebbace7e |
awk '{print $1}'
```

```
# sample outputs
sha256~-g-vg5YGGyYPp_TMKvcuGTatbwq6wklkrxY7ai49DYU
sha256~037FDmgjT5eTNVNSAeGzMAWqzsEvNeLItFAR_FLiEFQ
sha256~0vTcCQFaJDEZbVCBD_RpvOMA-ZO5yRIHAJH6DGEq3QY
...
```

4. Delete each of the token:

```
> oc delete oauthaccesstokens sha256~-g-
vg5YGGyYPp_TMKvcuGTatbwq6wklkrxY7ai49DYU

oauthaccesstoken.oauth.openshift.io "sha256~-g-
vg5YGGyYPp_TMKvcuGTatbwq6wklkrxY7ai49DYU" deleted

> oc delete oauthaccesstokens
sha256~037FDmgjT5eTNVNSAeGzMAWqzsEvNeLItFAR_FLiEFQ

oauthaccesstoken.oauth.openshift.io
"sha256~037FDmgjT5eTNVNSAeGzMAWqzsEvNeLItFAR_FLiEFQ" deleted

> oc delete oauthaccesstokens sha256~0vTcCQFaJDEZbVCBD_RpvOMA-
ZO5yRIHAJH6DGEq3QY

oauthaccesstoken.oauth.openshift.io "sha256~0vTcCQFaJDEZbVCBD_RpvOMA-
ZO5yRIHAJH6DGEq3QY" deleted
```

This step will revoke all the access tokens under the **`oauthclient`** name. In other words, it will also log out other users. This is due to there is no publicly known way to which token a certain user belong to in OpenShift Container Platform 4.6

.

"Must Gather" information for IBM Cloud Transformation Advisor

In all cases please get the following information:

1. The version of Transformation Advisor
2. The version of OpenShift that Transformation Advisor is installed in.

For issues related to the data collector

1. Version of the data collector (Version is in the extracted datacollector folder **`transformationadvisor-MAJOR.MINOR.PATCH`**)
2. Operating system type and version where the datacollector has been run
3. The type of collection being performed (e.g. WebSphere Application Server, JBoss etc)
4. The full command that was used to run the collector as well as any output to the console.
5. The Java version available on that system (If collecting from a Java application server)
6. If collection zip has been created (provided it is OK to share this data). The zip will contain the logs.
7. If unable to share the zip file, the logs from the log directory where the collector was run.

For issues where Transformation Advisor UI is not accessible and/or pods are not in a healthy state

1. Get the state of the pods:

   ```
   oc get pods -n <ta project>
   ```

2. Describe each pod

   ```
   oc describe pod <pod name> -n <ta project>
   ```

3. Get the logs for each pod

   ```
   oc logs <pod name> -n <ta project>
   ```

4. Get the route information

   ```
   oc get routes -n <ta project>
   ```

5. Get the persistence being used:

   ```
   oc get pvc -n <ta project>
   ```

   Get the PersistenceVolume status

   ```
   oc get pv <pvname>
   ```

where is shown under the VOLUME heading when you get the pvc in the previous command. For more on persistence, see Prerequisites.

For issues where UI is not functioning as expected

1. Browser type and version
2. The developer console output
   - To open the developer console in Google Chrome, open the Chrome menu in the browser and select **More Tools > Developer Tools**. You can also use the shortcut Option + ⌘ + J (on macOS), or Shift + CTRL + J (on Windows/Linux).
   - To open the developer console in Firefox, click on the Firefox menu in the browser and select **Web Developer > Browser Console**. You can also use the shortcut Shift + ⌘ + J (on macOS) or Shift + CTRL + J (on Windows/Linux).
   - Before you can access the developer console in Safari, you first need to enable the Developer Menu. To do that, go into Safari preferences (Safari Menu > Preferences) and select the **Advanced** tab. After that menu is enabled, you will find the developer console by clicking on **Develop > Show Javascript Console**. You can also use the shortcut Option + ⌘ + C.

.

What migration artifacts does Transformation Advisor create?

IBM Cloud Transformation Advisor produces migration artifacts for applications that are being migrated to Liberty on Red Hat ™ OpenShift 4.x. The migration artifacts kickstart the migration of your application. Transformation Advisor generates a Maven project for your application plus the resources that you need to build your application into an image and deploy on Red Hat ™ OpenShift.

The following table describes the artifacts that Transformation Advisor produces.

Artifact	Description
`server.xml`	Contains the configuration the Liberty server needs for your application. For example, if your application connects to a database, it includes a data source configuration.
`pom.xml`	The fundamental unit of work in Maven. It is used to build your application project if you have choosen a source project, or to pull in the application binary and dependencies if you have specified Maven coordinates for a binary project.
Source skeleton	If you specify a source project, you get a source skeleton project that contains a deployment descriptor and a simple `index.html` file.
Dockerfile	A multistage Dockerfile first builds the project. For a binary project specified with Maven coordinates, it pulls down the application binary and dependencies. Then, it creates a Liberty image ready for deployment.
Application CR	A custom resource (CR) configuration for your application. This resource will create an instance of your application from the Open Liberty Operator in Red Hat ™ OpenShift.

How to deploy your applications on Red Hat ™ OpenShift 4.x

The following describes how to use the Transformation Advisor migration artifacts to deploy your applications to a Liberty container and to an OpenShift cluster.

You have the option to:

- deploy a single application to a Liberty container.
- deploy multiple applications from a cluster or group to a Liberty container.

How to deploy a single application

A binary project

The following section describes how to deploy a binary based project to Red Hat™ OpenShift. For the binary project, you will provide binary files for the application and any dependencies.

You can migrate your application in three steps. This document will reference the following variables:

- **`<BINARY_FILE_LOCATION>`** is the location of the binary files for your application, including any dependencies and shared library files.
- **`<APPLICATION_FILE>`** is the binary file for your application.
- **`<DEPENDENCY_FILE_LOCATION>`** is the location of any dependencies and shared library files that your application requires.
- **`<APP_CONTEXT_ROOT>`** is the context root for your application. If not defined elsewhere, for example in ibm-web-ext-xml, this corresponds to the **`name`** attribute in the **`<application>`** element of the server.xml.
- **`<APPLICATION_NAME>`** is the name of the application.
- **`<CONTAINER_ID>`** is the container ID for your docker image. To get this value, enter: docker ps
- **`<IMAGE_REFERENCE>`** is the reference for this image, including the registry, repository and tag, e.g. docker.io/myspace/myappimage:1.0.0
- **`<LIBERTY_HOME>`** is the location where you have installed Liberty.
- **`<LIBERTY_MACHINE>`** is the machine where you have installed the Liberty profile.
- **`<MIGRATION_ARTIFACTS_HOME>`** is the location where you have unzipped the Transformation Advisor artifacts, or cloned the repository.
- **`<OCP_PROJECT>`** is the name of the OpenShift project where you want to install the application.

STEP ONE: Migrate the Java application to Liberty

In this step you will migrate your application to a local Liberty server. This will allow you to verify that your application works correctly on Liberty and make any configuration changes where necessary. After you have tested to make sure your application works on Liberty, you will be ready to create a Liberty container for your application.

Step 1 Prerequisites

- A Liberty installation

    - You can get WebSphere Liberty here: https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/twlp_inst.html
      Note: If your application requires Java EE6 features you will need Websphere Liberty
    - You can get Open Liberty here: https://openliberty.io/
- A copy of the migration artifacts that you downloaded from Transformation Advisor available on the machine where you have installed Liberty.

Step 1 Tasks

1. Add your application file to the migration bundle, and remove the placeholder file:

```
cp <BINARY_FILE_LOCATION>/<APPLICATION_FILE> <MIGRATION_ARTIFACTS_HOME>/target/
rm <MIGRATION_ARTIFACTS_HOME>/target/*.placeholder
```

2. Add your dependency file(s) to the migration bundle, and remove the placeholder file(s):

```
cp <DEPENDENCY_FILE_LOCATION>/* <MIGRATION_ARTIFACTS_HOME>/src/main/liberty/lib
rm <MIGRATION_ARTIFACTS_HOME>/src/main/liberty/lib/*.placeholder
```

```
 NOTE: The placeholder files provide the name(s) of all depednecies for your
application
```

3. Create a server in the Liberty installation to run your application:

```
cd <LIBERTY_HOME>/bin
./server create server1
```

4. Go to the location of your migration artifacts and copy the application binary (ear/war) in the target directory to the apps directory of Liberty:

```
cd <MIGRATION_ARTIFACTS_HOME>
cp target/*.war <LIBERTY_HOME>/usr/servers/server1/apps
```

5. If it doesn't already exist, create the directory

```
<LIBERTY_HOME>/usr/shared/config/lib/global
```

and copy any additional binaries from the migration location to that location:

```
mkdir -p <LIBERTY_HOME>/usr/shared/config/lib/global
cp src/main/liberty/lib/* <LIBERTY_HOME>/usr/shared/config/lib/global
```

6. Update the server.xml if necessary:
    - When Transformation Advisor creates the server.xml file for the migration bundle, sensitive data is extracted as variables at the end of the `server.xml` file with an empty default values. Ensure the variable default values are set appropriately.
    - If there are any additional binaries listed in this file that you do not need, remove any reference to them.

NOTE: Only set default value for the variables defined in the server.xml file. If the value is set in the server.xml file, they cannot be overrode during the deployment time.

7. Copy the generated server.xml into place, replacing the default server.xml:

```
cp src/main/liberty/config/server.xml
<LIBERTY_HOME>/usr/servers/server1/server.xml
```

8. Start the Liberty server:

```
<LIBERTY_HOME>/bin/server start server1
```

9. Check the Liberty logs to confirm that your application has started correctly and to find the URL to for access:

```
cd <LIBERTY_HOME>/usr/servers/server1/logs
vi messages.log
```

NOTE: You may have defined features in the **`server.xml`** that are not installed by default in Liberty. These will be listed in the log. In this case install the necessary features using **`bin/featureUtility`**

NOTE: If you define a **`<dataSource>`** element in the server.xml, you may encounter an authentication issue similar to this: **`invalid username/password; logon denied`**

If you see this issue, you may need to update the default value for the auth data variables used in your data source properties.

10. Verify that the logs contain a line similar to this: TCP Channel defaultHttpEndpoint has been started and is now listening for requests on host * (IPv6) port 9080 If you do not see this then there has been some problem starting the server or launching the application. Search through the log file for more details and debug accordingly.
11. Open your application in the browser by going to the following link: **`http://<LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT_ROOT>`**

NOTE: The migration artifacts assist you in the migration of your application. Depending on the nature and complexity of your application, additional configuration may be required to fully complete this task. For example, you may need to complete extra configuration to connect your application with a user registry, and or to configure user security role bindings. Consult the product documentation for more details.

STEP TWO: Containerize Liberty

In this step you will containerize your working Liberty installation. You will create a Liberty image that has your migrated application installed and working, and then test the image to confirm that it is operating correctly.

`NOTE: If you are using podman instead of docker simply replace the word docker in each command with podman`

Step 2 Prerequisites

- You have completed Step 1: Migrate the Java application to Liberty
- Docker or podman is installed.
  - Download Docker: https://www.docker.com/get-started
  - Download podman: https://podman.io/getting-started/installation
- The machine where you complete this task requires access to the internet to download the Liberty base image.

Step 2 Tasks

1. Stop the Liberty server if it is running to ensure that the necessary ports are available:

```
<LIBERTY_HOME>/bin/server stop server1
```

2. If using docker ensure the docker service is running. If it?s not, start it:

```
service docker start
```

3. Go to where your migration artifacts are located and build your image from the docker file:

```
cd <MIGRATION_ARTIFACTS_HOME>
docker build --no-cache -t "<IMAGE_REFERENCE>" -f Containerfile .
```

Note: The migration bundle includes a pom.xml file that allows the application and all dependencies to be pulled from Maven. You need to enable this option in the Containerfile and update the placeholders in the pom.xml file with correct values. The details of how to do this can be found in the 'Add dependencies using Maven' section

4. Run the image and confirm that it is working correctly:

```
docker run -p 9080:9080 <IMAGE_REFERENCE>
```

5. If everything looks good, the image has been started and mapped to the port 9080. You can access it from your browser with this link: `http://<LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT_ROOT>` **Optional:** Check your image when it is up and running by logging into the container:

```
docker exec -ti <CONTAINER_ID> bash
```

This allows you to browse the file system of the container where your application is running.

STEP THREE: Deploy your image to Red Hat OpenShift

In this step you will deploy the image you have created to Red Hat OpenShift. These instructions relate to OpenShift 4+ and have been validated on OpenShift 4.12.

Step 3 Prerequisites

- Access to either a public or private Red Hat OpenShift 4+ environment.
- Image registry access
  - You will need push your migrated application image to a location that is accessible to the OpenShift cluster. You may use a publicly available registry (e.g. Dockerhub), or your own private registry. The migration artifacts generated by Transformation Advisor (specifically the application-cr.yaml file) need to be updated with the image reference that you are using. If you do not have a suitable registry available you can create your own in Dockerhub or Podman to use until a suitable registry is found.
    - How to create your own registry in Dockerhub can be found here
    - How to create your own registry in Podman can be found here
- A copy of the migration artifacts that you downloaded from Transformation Advisor available on the machine where you have installed Liberty.
- WebSphere Liberty Operator or Open Liberty Operator
  - In order to deploy your application on WebSphere Liberty on OpenShift, you must first have installed the WebSphere Liberty operator on your cluster.
    - The WebSphere Liberty Operator is available in the IBM Operator Catalog. Go to the Operators...OperatorHub UI and click the **IBM Operator Catalog** `Source`. You can use the search to quickly find the WebSphere Liberty operator.
    - Click on the WebSphere Liberty operator UI and follow the instructions to install.
    - You will be given the option of installing the operator cluster wide which will be capable of installing WebSphere Liberty applications across all namespaces, OR, you can choose to install the operator in a given namespace. Choose either option depending on your preference.
    - For full details on the WebSphere Liberty operator, please consult the documentation here.
    - For details on how to install the IBM Operator Catalog, please consult the documentation here.

- In order to get support for your application deployed on Open Liberty on OpenShift, you need to use WebSphere Liberty operator. But you can still switch to use the Open Liberty operator to do the deployment.
  - you must first have installed the Open Liberty operator on your cluster.
  - The Open Liberty Operator is available in the Certified Catalog. Go to the Operators...OperatorHub UI and click the **Certified** `Source`. You can use the search to quickly find the Open Liberty operator.
  - Click on the Open Liberty operator UI and follow the instructions to install.
  - You will be given the option of installing the operator cluster wide which will be capable of installing Open Liberty applications across all namespaces, OR, you can choose to install the operator in a given namespace. Choose either option depending on your preference.
  - For full details on the Open Liberty operator, please consult the documentation here

Step 3 Tasks

1. Push your application image to a registry that is accessible to the OpenShift cluster. e.g.

   ```
   docker push <IMAGE_REFERENCE>
   ```

   where **`<IMAGE_REFERENCE>`** is the image reference you tagged the image with when building, e.g. **`docker.io/myspace/myappimage:1.0.0`**

   Note:

   - You may need to log in to that image registry in order to successfully push.
   - If your registry requires authenticated pulls, you will need to set up your cluster with a pull secret. See docs for more information.
2. Log in to the OpenShift cluster and create a new project in OpenShift.

   ```
   oc login -u <USER_NAME>
   ```

   If you have not already created a namespace when installing the Liberty operator create one now.

   ```
   oc new-project <OCP_PROJECT>
   ```

3. Deploy your image with its accompanying operator using the following instructions:

   a. Change to the kustomize directory in **`<MIGRATION_ARTIFACTS_HOME>`**:

   ```
   cd <MIGRATION_ARTIFACTS_HOME>/deploy/kustomize
   ```

   b. Update the IMAGE_REFERENCE in the **`base/application-cr.yaml`** file to match the application image reference you pushed in Task 1.

   c. Check the value defined in the **`base/<APPLICATION_NAME>-configmap.yaml`** file are all correct.

   d. Update the **`overlays/dev/<APPLICATION_NAME>-secret.yaml`** file, to add the value for the sensitive date. The value need to be base64 encoded. You can invoke command **`echo -n 'your-secret-password' | base64`** to get the encoded string for your sensitive data.

   e. Create the custom resource (CR) for your Liberty application using **`apply -k`** command to specify a directory containing kustomization.yaml:

   ```
   oc apply -k overlays/dev
   ```

   f. You can view the status of your deployment by running oc get deployments. If you don?t see the status after a few minutes, query the pods and then fetch the Liberty pod logs:

```
oc get pods
oc logs <pod>
```

g. You can now access your application by navigating to the **`Networking...Routes`** UI in OpenShift and selecting the namespace where you deployed.

4. **Optional:** If you wish to delete your application from OpenShift, you can uninstall using the OpenShift UI. Navigate to the **`Operators...Installed operators`** in OpenShift to Find the Liberty operator. Click on the **`All instances`** tab to find your application instance and uninstall as required.

Add dependencies using Maven

Overview

The migration bundle includes a pom.xml file that allows the application and all dependencies to be pulled from Maven. You need to enable this option in the Containerfile and update the placeholders in the pom.xml file with correct values.

Tasks

1. Navigate to the migration bundle

```
cd <MIGRATION_ARTIFACTS_HOME>
```

2. Edit the pom.xml file and update the placeholder values with the appropriate values
    1. Set the correct values for the **`<dependency>`** element
    2. Set the correct values for each of the **`<artifactItem>`** elements
3. Edit the Containerfile so that dependencies are pulled in during the image build
    1. Uncomment the line **`RUN mvn -X initialize process-resources verify`**

A source project

Overview

When migrating an application, you will often need to make changes to the source code to ensure a successful migration to the new target platform. The exact nature of the changes will vary from application to application. Transformation Advisor reports on the changes necessary for each individual application and will classify applications that require code changes as either Moderate or Complex. To pinpoint exactly where these changes need be made in the code you can use the WebSphere Application Migration Toolkit (WAMT) Eclipse plugin. The tool can also suggest possible fixes. See this link for more details:
https://developer.ibm.com/wasdev/downloads/#asset/tools-WebSphere_Application_Server_Migration_Toolkit
The following tasks help you build your source code to an image.

Tasks

1. Navigate to the migration bundle

```
cd <MIGRATION_ARTIFACTS_HOME>
```

2. Edit the pom.xml file and update or delete the placeholder values with the appropriate values
    1. Set the correct values for the **`<dependency>`** element
    2. Set the correct values for each of the **`<artifactItem>`** elements
3. Update the pom.xml file to build the application according to your specifications

    1. NOTE: If you already have a pom.xml for your application then you can use that, or merge with the migration bundle pom.xml as appropriate
4. Edit the Containerfile so that source code will be built during the image build
    1. Uncomment the line `RUN mvn clean package`

How to deploy a cluster or group of applications

This section covers how to use the migration bundle for a cluster or group to deploy multiple applications to a single Liberty container.
You can migrate each application individually first and then merge them into a single deployment, or you can deploy them all at once as a single first step.

If you are migrating a cluster then it is unlikely that you will get a clash of features between your applications. In this case the "All applications at once approach" will be the fastest approach to migrate your cluster.

If you are migrating a group then it is more likely that you may get a feature clash when deploying the applications together, because the applications may not have run together before. It is recommended to take the "All applications at once approach" but if you encounter issues, it is recommended to take the "Application by application" approach to resolve them.

All applications at once

You can migrate all your applications in three steps. This document will reference the following variables:

- `<APP_CONTEXT_ROOT>` is the context root for your application. If not defined elsewhere, for example in ibm-web-ext-xml, this corresponds to the `name` attribute in the `<application>` element of the server.xml.
- `<APPLICATION_NAME>` is the name of the application.
- `<MIGRATION_ARTIFACTS_HOME>` is the location where you have unzipped the Transformation Advisor artifacts, or cloned the repository.
- `<IMAGE_REFERENCE>` is the reference for this image, including the registry, repository and tag, e.g. docker.io/myspace/myappimage:1.0.0

STEP ONE: Gather applications and dependency files and update configuration

In this step you will gather your application files and any dependencies and update the configuration files.

Step 1 Tasks

1. Add your application file(s) to the migration bundle, and remove the placeholder files:

```
cp <BINARY_FILE_LOCATION>/<APPLICATION_FILE>
<MIGRATION_ARTIFACTS_HOME>/target/
rm <MIGRATION_ARTIFACTS_HOME>/target/*.placeholder
```

2. Add your dependency file(s) to the migration bundle, and remove the placeholder file(s):

```
cp <DEPENDENCY_FILE_LOCATION>/*
<MIGRATION_ARTIFACTS_HOME>/src/main/liberty/lib
rm <MIGRATION_ARTIFACTS_HOME>/src/main/liberty/lib/*.placeholder
```

```
NOTE: The placeholder files provide the name(s) of all dependencies for your
application
```

NOTE: If you are using maven to import your application and dependencies you can skip task 1 & 2 of this step

3. Update any application server.xml files if necessary:
   - The server.xml file at **<MIGRATION_ARTIFACTS_HOME>/src/main/liberty/config** contains a series of includes, that include the server.xml files for each application
   - The application server.xml files can be found in the following locations: **<MIGRATION_ARTIFACTS_HOME>/apps/<APPLICATION_NAME>/src/main/liberty/config**
   - The name of the server.xml file for each application will be **<APPLICATION_NAME>_server_config.xml**
   - Modify each server.xml files by entering default values for any sensitive data that Transformation Advisor has removed.
   - If there are any additional binaries listed in the server.xml file that you do not need, remove any reference to them.

NOTE: Only set default value for the variables defined in the server.xml file. If the value is set in the server.xml file, they cannot be overrode during the deployment time.

STEP TWO: Containerize all applications on Liberty

In this step you will containerize your working Liberty installation. You will create a Liberty image that has all your migrated applications installed and working, and then test the image to confirm that it is operating correctly.

NOTE: If you are using podman instead of docker simply replace the word docker in each command with podman

Step 2 Prerequisites

- You have completed Step 1: Gather applications and dependency files and update configuration
- Docker or podman is installed.
  - Download Docker: https://www.docker.com/get-started
  - Download podman: https://podman.io/getting-started/installation
- The machine where you complete this task requires access to the internet to download the Liberty base image.

Step 2 Tasks

1. If using docker ensure the docker service is running. If it?s not, start it:

```
service docker start
```

2. Go to where your migration artifacts are located and build your image from the docker file:

```
cd <MIGRATION_ARTIFACTS_HOME>
docker build --no-cache -t "<IMAGE_REFERENCE>" -f Containerfile .
```

Note: The migration bundle includes a pom.xml file that allows the application and all dependencies to be pulled from Maven. You need to enable this option in the Containerfile and update the placeholders in the pom.xml file with correct values. The details of how to do this can be found in the 'Add dependencies using Maven' section.

4. Run the image and confirm that it is working correctly:

```
docker run -p 9080:9080 <IMAGE_REFERENCE>
```

5. If everything looks good, the image has been started and mapped to the port 9080.
   You can access it from your browser with this link: **`http://<LIBERTY_HOME_MACHINE_IP>:9080/>`**

6. Each application will be available at
   `http://<LIBERTY_HOME_MACHINE_IP>:9080/<APP_CONTEXT_ROOT>`

7. **Optional:** Check your image when it is up and running by logging into the container:

```
docker exec -ti <CONTAINER_ID> bash
```

This allows you to browse the file system of the container where your applications are running.

STEP THREE: Deploy your image with all applications to Red Hat OpenShift

In this step you will deploy the image you have created to Red Hat OpenShift. These instructions relate to OpenShift 4+ and have been validated on OpenShift 4.12. Follow the same steps as if your image had a single application - see here

A binary project - application by application

In this approach you will configure, containerize and deploy each application individually, and then deploy them together

Tasks

1. Complete the steps for 'How to deploy a single application' for each application.

```
cd <MIGRATION_ARTIFACTS_HOME>/apps
Deploy each application indvidually
```

2. Now deploy all applications together following these steps

NOTE: You may have a feature conflict when you deploy applications together. In this case you need to either move the apps that are causing the conflict into another deployment, or update the applications so they no longer require conflicting features.

Managing keystores during deployment

```
When deploying to OpenShift, if your application is configured to use non default
keystores then the default route will not work without configuration changes
```

Configuring keystores in a container

NOTE: If you plan to deploy to OpenShift you can use the Operator to generate the necessary certificates, see the next section for details.

Transformation Advisor does not automatically migrate keystore information in the migration bundle.
When running in a container using the default Containerfile produced by Transformation Advisor the Liberty server will output messages indicating that the non-default keystore files can not be found.

For instructions on configuring keystores in Liberty server, see the Liberty Configuring Security documentation

Configuring keystores in an OpenShift deployment

When deploying to OpenShift you can use the Liberty Operator to generate all necessary certificates or use your own certificates.

Using the Liberty Operator to generate certificates

Complete the following steps to use the Liberty Operator to generate the certificates for your application

1. Modify the `server.xml` file and remove all `<keystore>` attributes.
2. Build and deploy the image as normal

In this case all certification generation and keystore management will be handled by the Operator. Further details on this can be found in the Liberty documentation

Using your own certificate and keystores in OpenShift

You can configure your deployment to use your own certificates and manage the security yourself. Further details on this can be found in the Liberty documentation for specifying certificates

.

How to configure your deployed application

By default, your application automatically externalizes variables that can be uniquely configured for each deployment. At runtime, the values for these variables are read from the environment variables set in the container. If no corresponding environment variable is found, the default value that is specified in the `server.xml` file is used. However, if the variable represents sensitive data (such as passwords), no default value is provided in the `server.xml` file. In this case, the variable must be defined as an environment variable for the deployment to work properly.

Default variables

The default value that is defined for each variable matches the value that is collected during analysis.

An example of how variables are used in the `server.xml` file:

```
<httpEndpoint host="${httpEndpoint_host_1}"
httpsPort="${httpEndpoint_secure_port_1}" id="defaultId"/>
```

An example of the default values set for these variables:

```
<variable defaultValue="*" name="httpEndpoint_host_1"/>
<variable defaultValue="9443" name="httpEndpoint_secure_port_1"/>
```

Sensitive data variables

Sensitive data values must be provided through environment variables.

An example of how sensitive data variables are used in the `server.xml` file:

```
<authData id="whydah1Node04/db2" password="${whydah1Node04_db2_password_1}"
user="user1"/>
```

No default values are defined for sensitive data variables.

File-based transaction logging

NOTE: If your application is configured to use file-based transaction logging, it requires configuration changes to run in a container. Either remove the file-based transaction logging from the **`server.xml`** file or configure the container to have the appropriate storage.

Removing file-based transaction logging

You need to remove the **`<transaction />`** element from the **`server.xml`** file.

1. **`cd <MIGRATION_BUNDLE_HOME>/src/main/liberty/config`**
2. Remove the **`<transaction />`** element from the **`server.xml`** file.

NOTE: Your deployed application no longer has transaction logging or recovery.

Configuring file-based transaction logging in a container

When you deploy your application to containers, ensure that the transaction log directory path is located within a mounted external volume. If you deploy the application to Kubernetes, use a Persistent Volume that supports Read-Write Many access and does not cross data center boundaries.
For more information, see Transaction recovery in a cloud environment.

Memory Based Session Replication

NOTE: Memory-based session replication requires configuration changes to work properly in a container environment.

Configuring memory-based Session replication in a container

Two solutions provide equivalent functionality for configuring memory-based session replication in a containerized application.

Solution 1: Session Persistence with JCache (Liberty)

For equivalent functionality in Liberty, you can configure session caching by using JCache with Red Hat Data Grid, based on Infinispan. Alternatively, you can use Hazelcast or other JCache providers.

For instructions on configuring containers based on an official Liberty container image to use Red Hat Data Grid (Infinispan) or Hazelcast, see session caching in Open Liberty Images.

For instructions on enabling session caching in any Liberty server, including containers not based on an official Liberty container image, see Configuring Liberty session persistence with JCache.

Solution 2: Session Persistence to a database (Liberty or WebSphere Application Server traditional (base))

An alternative solution to using JCache is to persist session data to a database.
The database solution is available for both Liberty and WebSphere traditional base:

- For Liberty, see Configuring Liberty session persistence to a database
- For WebSphere Application Server traditional (base), see Configuring for database session persistence

Getting help for IBM Cloud Transformation Advisor

If you encounter any issues with IBM Cloud Transformation Advisor, you can find assistance in several ways.

Finding answers to your questions

- Post your questions on StackOverflow for IBM Cloud Transformation Advisor.
- Search for answers to your questions using the *IBM Documentation* search field. Explore the search results on the *Documentation*, *Videos*, *IBM Developer*, *Technotes*, and *Redbooks* tabs.

Reporting issues

- If you want to report a problem or are looking for support options for IBM Cloud Transformation Advisor, see the Support Portal for one of the following products, depending on your license:
  - IBM Cloud Pak for Applications
  - IBM Cloud Pak for Integration
  - IBM WebSphere Hybrid Edition
  - WebSphere Application Server for z/OS
  - WebSphere Automation

Contacting sales

Contact sales with your queries.

1. Go to the one of the following web sites:
   - IBM Cloud Pak for Applications
   - IBM Cloud Pak for Integration
   - IB< WebSphere Hybrid Edition
   - WebSphere Application Server for Z/OS
   - WebSphere Automation
2. Click *Let's talk*.
3. Select to *Call sales*, *Email sales*, *Book a meeting* to schedule a consulation with an IBM expert, or *Chat with sales*.

Support for IBM Cloud Transformation Advisor

Learn how to get notifications, support, and information about support lifecycle policies for IBM® Cloud Transformation Advisor.

Signing up for notifications about IBM Cloud Transformation Advisor

You can be informed of critical IBM software support updates by using the My Notifications subscription service. For more information, see Stay up to date with My Notifications on the IBM website.

To subscribe to products of your choice, see My Notifications.

Opening a support case with IBM

Before you contact IBM Support, check the Day 2 and troubleshooting documentation.

To learn more about IBM Support, see the IBM Support Guide and the Getting Started Guide for IBM Support.

To open a support ticket, you must have an active entitlement for Mono2Micro. If you do not have an active entitlement, see Licenses under which you have entitlement to use Transformation Advisor for a list of offerings.

Before you open a support case, if you are running in a cluster, collect MustGather data about your cluster.

To open a support case with IBM, follow these steps.

1. Go to the IBM Support site.
2. From the menu bar on the header, click *Open a case*.
3. Log in with your IBMid and password.
4. Enter a meaningful *Case Title* that summarizes your problem.
5. Select IBM as the *Product Manufacturer*.
6. Select the Product through which you have a license to use IBM Cloud Transformation Advisor. Depending on your license, the product is one of the following:
   - IBM Cloud Pak for Applications
   - IBM Cloud Pak for Integration
   - IBM WebSphere Hybrid Edition
   - WebSphere Application Server for z/OS.
   - WebSphere Automation
7. Select the appropriate Severity of the problem. For more information about problem severity, see IBM Enterprise Support Severity Definitions.

   Note: The case severity is based on the business impact of the problem. If you set the case severity as 1, you must be available 24 hours a day to work with IBM Support on the issue.
8. Select the Account that has the entitlement for IBM Cloud Transformation Advisor.
9. Provide a detailed Case Description of your problem. A detailed description can help support understand your problem more accurately and thus provide quicker solutions or answers. The following information is crucial:
   - IBM Cloud Transformation Advisor product version
   - Installation platform (VMware, Azure, AWS, IBM Cloud)
   - Red Hat OpenShift Container Platform version
   - Steps to reproduce the issue
10. Collect information about your cluster. For more information about how to gather the required information, see "Must Gather" information for IBM Cloud Transformation Advisor.
11. Upload the tar.gz file with the results of the diagnostic scan that you ran in the previous step in one of the following ways:
    - Upload a file during the case creation

- Send large files to Enhanced Customer Data Repository (ECuRep)

12. Select the language preferences.

13. Click Submit a case.

Note: Open a case for each problem that you need assistance with. Do not add new issues to an existing case for which you are already engaged with the support team. Clearly and completely define the issue to reduce confusion for the support team.

Understanding support lifecycle policies

IBM Cloud Transformation Advisor follows a modified IBM Continuous Delivery (CD) Software Support Lifecycle Policy.

The numbering scheme for the version of WebSphere Automation is based on the semantic versioning specification defined at: http://semver.org/.

The version numbering takes the form *major.minor.patch*, where:

MAJOR Number changes occur when significant changes are introduced between releases, including incompatible changes. *MINOR* Number changes indicate a new release. *PATCH* Number changes for fix updates.

IBM Cloud Transformation Advisor treats each major version (as in major.minor.patch semantic versioning) as a new IBM CD Release with new support dates. New minor and patch versions are both equivalent to an IBM "CD update package".

IBM Support is provided for the current major version (as in major.minor.patch semantic versioning) and the minor versions for the previous major version that are less than two years old

Fixes and new functionality will only be provided in the next CD update package.

For supported open source components of IBM Cloud Transformation Advisor:

```
If an identified problem requires a fix to an open source project, the ability to
deliver this fix is contingent on the open source community project accepting and
publishing the fix for the required version of the open source project.  IBM will
supply reasonable effort to identify the fix and work with the community to have
the fix included in an appropriate release.
```

For more information about IBM software lifecycle policies, see https://www.ibm.com/support/pages/node/718165.

Resources Related to IBM Cloud Transformation Advisor and Application Modernization

Try Transformation Advisor

See the IBM Garage site and try a free 180-day trial of Transformation Advisor on your laptop (requires Docker Desktop).

Modernization Playbooks

- Interactive Java Application Modernization Playbook
- IBM's Modernization Playbook

IBM Application Product Information

- IBM Application Product Information and Guides

Blog on Modernization

- Check out the blog on the difference between application migration and modernization.

Demo and Sample Code

- View a demo of migrating on-premises applications to the cloud with this sample code pattern. The code features a sample web application, including source code with detailed instructions and a video.