Migration Toolkit for Applications 7.0

User Interface Guide

Use the Migration Toolkit for Applications user interface to group your applications into projects for analysis.

Last Updated: 2024-05-03

Migration Toolkit for Applications 7.0 User Interface Guide

Use the Migration Toolkit for Applications user interface to group your applications into projects for analysis.

Legal Notice

Abstract

This guide describes how to use the Migration Toolkit for Applications user interface to accelerate large-scale application modernization efforts across hybrid cloud environments on Red Hat OpenShift.

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright's message .

CHAPTER 1. INTRODUCTION

1.1. ABOUT THE USER INTERFACE GUIDE

This guide is for architects, engineers, consultants, and others who want to use the Migration Toolkit for Applications (MTA) user interface to accelerate large-scale application modernization efforts across hybrid cloud environments on Red Hat OpenShift. This solution provides insight throughout the adoption process, at both the portfolio and application levels: inventory, assess, analyze, and manage applications for faster migration to OpenShift via the user interface.

NOTE

The migration solution that was provided in the Migration Toolkit for Applications 5.*x* releases (migration and modernization of Java applications) is now available with Migration Toolkit for Runtimes 1.0.

1.2. ABOUT THE MIGRATION TOOLKIT FOR APPLICATIONS

What is the Migration Toolkit for Applications?
Migration Toolkit for Applications (MTA) accelerates large-scale application modernization efforts across hybrid cloud environments on Red Hat OpenShift. This solution provides insight throughout the adoption process, at both the portfolio and application levels: inventory, assess, analyze, and manage applications for faster migration to OpenShift via the user interface.

MTA uses an extensive default questionnaire as the basis for assessing your applications, or you can create your own custom questionnaire, enabling you to estimate the difficulty, time, and other resources needed to prepare an application for containerization. You can use the results of an assessment as the basis for discussions between stakeholders to determine which applications are good candidates for containerization, which require significant work first, and which are not suitable for containerization.

MTA analyzes applications by applying one or more rulesets to each application considered to determine which specific lines of that application must be modified before it can be modernized.

MTA examines application artifacts, including project source directories and application archives, and then produces an HTML report highlighting areas needing changes.

How does the Migration Toolkit for Applications simplify migration?
The Migration Toolkit for Applications looks for common resources and known trouble spots when migrating applications. It provides a high-level view of the technologies used by the application.

MTA generates a detailed report evaluating a migration or modernization path. This report can help you to estimate the effort required for large-scale projects and to reduce the work involved.

1.3. ABOUT THE USER INTERFACE

The user interface for the Migration Toolkit for Applications allows a team of users to assess and analyze applications for risks and suitability for migration to hybrid cloud environments on Red Hat OpenShift.

Use the user interface to assess and analyze your applications to get insights about potential pitfalls in the adoption process, at both the portfolio and application levels as you inventory, assess, analyze, and manage applications for faster migration to OpenShift.

CHAPTER 2. USER INTERFACE VIEWS

The Migration Toolkit for Applications (MTA) user interface has two views:

- Administration view
- Migration view

In **Administration** view, you configure the instance environment, working with credentials, repositories, HTTP and HTTPS proxy definitions, custom migration targets, and issue management.

In **Migration** view, you perform application assessments and analyses, review reports, and add applications for assessment and analysis.

CHAPTER 3. INSTALLING THE MIGRATION TOOLKIT FOR APPLICATIONS USER INTERFACE

You can install the Migration Toolkit for Applications (MTA) user interface as part of the process of installing the MTA Operator on the OpenShift Container Platform.

The MTA Operator is a structural layer that manages resources deployed on Kubernetes (database, front end, back end) to automatically create an MTA instance.

3.1. PERSISTENT VOLUME REQUIREMENTS

To successfully deploy, the MTA Operator requires 3 RWO persistent volumes (PVs) used by different components. If the **rwx_supported** configuration option is set to **true**, the MTA Operator requires an additional 2 RWX PVs that are used by Maven and the hub file storage. The PVs are described in the table below:

Table 3.1. Required persistent volumes

Name	Default size	Access mode	Description
hub database	10 GiB	RWO	Hub database
hub bucket	100 GiB	RWX	Hub file storage; required if the **rwx_supported** configuration option is set to **true**
keycloak postgresql	1 GiB	RWO	Keycloak back end database
pathfinder postgresql	1 GiB	RWO	Pathfinder back end database
cache	100 GiB	RWX	Maven m2 cache; required if the **rwx_supported** configuration option is set to **true**

3.2. INSTALLING THE MIGRATION TOOLKIT FOR APPLICATIONS OPERATOR AND THE USER INTERFACE

You can install the Migration Toolkit for Applications (MTA) and the user interface on OpenShift Container Platform versions 4.13-4.15 when you install the Migration Toolkit for Applications Operator.

Prerequisites

- 4 vCPUs, 8 GiB RAM, and 40 GiB persistent storage.
- OpenShift Container Platform 4.13-4.15 installed.

- You must be logged in as a user with **cluster-admin** permissions.

For more information, see OpenShift Operator Life Cycles.

Procedure

1. In the OpenShift Container Platform web console, click **Operators → OperatorHub**.
2. Use the **Filter by keyword** field to search for **MTA**.
3. Click the **Migration Toolkit for Applications** Operator and then click **Install**.
4. On the **Install Operator** page, click **Install**.
5. Click **Operators → Installed Operators** to verify that the MTA Operator appears in the **openshift-mta** project with the status **Succeeded**.
6. Click the **MTA** Operator.
7. Under **Provided APIs**, locate **Tackle**, and click **Create Instance**.
   The **Create Tackle** window opens in **Form** view.
8. Review the custom resource (CR) settings. The default choices should be acceptable, but make sure to check the system requirements for storage, memory, and cores.
9. To work directly with the YAML file, click **YAML** view and review the CR settings that are listed in the **spec** section of the YAML file.
   The most commonly used CR settings are listed in this table:

Table 3.2. Tackle CR settings

Name	Default	Description
cache_data_volume_size	**100 GiB**	Size requested for the cache volume; ignored when **rwx_supported=false**
cache_storage_class	Default storage class	Storage class used for the cache volume; ignored when **rwx_supported=false**
feature_auth_required	**True**	Flag to indicate whether keycloak authorization is required (single user/“noauth”)
feature_isolate_namespace	**True**	Flag to indicate whether namespace isolation using network policies is enabled
hub_database_volume_size	**10 GiB**	Size requested for the Hub database volume
hub_bucket_volume_size	**100 GiB**	Size requested for the Hub bucket volume
hub_bucket_storage_class	Default storage class	Storage class used for the bucket volume

Name	Default	Description
keycloak_database_data_volume_size	**1 GiB**	Size requested for the Keycloak database volume
pathfinder_database_data_volume_size	**1 GiB**	Size requested for the Pathfinder database volume
maven_data_volume_size	**100 GiB**	Size requested for the Maven m2 cache volume; deprecated in MTA 6.0.1
rwx_storage_class	NA	Storage class requested for the Tackle RWX volumes; deprecated in MTA 6.0.1
rwx_supported	**True**	Flag to indicate whether the cluster storage supports RWX mode
rwo_storage_class	NA	Storage class requested for the Tackle RW0 volumes
rhsso_external_access	**False**	Flag to indicate whether a dedicated route is created to access the MTA managed RHSSO instance
analyzer_container_limits_cpu	**1**	Maximum number of CPUs the pod is allowed to use
analyzer_container_limits_memory	**4GiB**	Maximum amount of memory the pod is allowed to use. You can increase this limit if the pod displays **OOMKilled** errors.
analyzer_container_requests_cpu	**1**	Minimum number of CPUs the pod needs to run
analyzer_container_requests_memory	**4GiB**	Minimum amount of memory the pod needs to run

Example YAML file

```
kind: Tackle
apiVersion: tackle.konveyor.io/v1alpha1
metadata:
  name: mta
  namespace: openshift-mta
spec:
  hub_bucket_volume_size: "25Gi"
  maven_data_volume_size: "25Gi"
  rwx_supported: "false"
```

10. Edit the CR settings if needed, and then click **Create**.

11. In **Administration** view, click **Workloads → Pods** to verify that the MTA pods are running.

12. Access the user interface from your browser by using the route exposed by the **mta-ui** application within OpenShift.

13. Use the following credentials to log in:

    - **User name**: admin

    - **Password**: Passw0rd!

14. When prompted, create a new password.

3.3. INSTALLING THE MIGRATION TOOLKIT FOR APPLICATIONS OPERATOR IN A DISCONNECTED OPENSHIFT CONTAINER PLATFORM ENVIRONMENT

You can install the MTA Operator in a disconnected environment by following the instructions in generic procedure.

In step 1 of the generic procedure, configure the image set for mirroring as follows:

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: registry.to.mirror.to
    skipTLS: false
mirror:
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.15
    packages:
    - name: mta-operator
      channels:
      - name: stable-v7.0
    - name: rhsso-operator
      channels:
      - name: stable
  helm: {}
```

3.4. MEMORY REQUIREMENTS FOR RUNNING MTA ON RED HAT OPENSHIFT LOCAL

When installed on Red Hat OpenShift Local, MTA requires a minimum amount of memory to complete its analysis. Adding memory makes the analysis process run faster. The table below describes the MTA performance with varying amounts of memory.

Table 3.3. OpenShift Local MTA memory requirements

Memory (GiB)	Description
10	MTA cannot run the analysis due to insufficient memory

Memory (GiB)	Description
11	MTA cannot run the analysis due to insufficient memory
12	**MTA works and the analysis is completed in approximately 3 minutes**
15	MTA works and the analysis is completed in less than 2 minutes
20	MTA works quickly, and the analysis is completed in less than 1 minute

The test results indicate that the minimum amount of memory for running MTA on OpenShift Local is **12 GiB**.

NOTE

- The tests were performed by running the MTA binary analysis through the user interface.
- All the analyses used the **tackle-testapp** binary.
- All the tests were conducted on an OpenShift Local cluster without the monitoring tools installed.
- Installing the cluster monitoring tools requires an additional 5 GiB of memory.

3.4.1. Eviction threshold

Each node has a certain amount of memory allocated to it. Some of that memory is reserved for system services. The rest of the memory is intended for running pods. If the pods use more than their allocated amount of memory, an out-of-memory event is triggered and the node is terminated with a **OOMKilled** error.

To prevent out-of-memory events and protect nodes, use the **--eviction-hard** setting. This setting specifies the threshold of memory availability below which the node evicts pods. The value of the setting can be absolute or a percentage.

Example of node memory allocation settings

- Node capacity: **32 GiB**
- **--system-reserved** setting: **3 GiB**
- **--eviction-hard** setting: **100 MiB**

The amount of memory available for running pods on this node is 28.9 GiB. This amount is calculated by subtracting the **system-reserved** and **eviction-hard** values from the overall capacity of the node. If the memory usage exceeds this amount, the node starts evicting pods.

3.5. RED HAT SINGLE SIGN-ON

MTA delegates authentication and authorization to a Red Hat Single Sign-On (RHSSO) instance managed by the MTA operator. Aside from controlling the full lifecycle of the managed RHSSO instance, the MTA operator also manages the configuration of a dedicated realm that contains all the roles and permissions that MTA requires.

If an advanced configuration is required in the MTA managed RHSSO instance, such as adding a provider for User Federation or integrating identity providers, users can log into the RHSSO Admin Console through the /**auth**/**admin** subpath in the **mta-ui** route. The admin credentials to access the MTA managed RHSSO instance can be retrieved from the **credential-mta-rhsso** secret available in the namespace in which the user interface was installed.

A dedicated route for the MTA managed RHSSO instance can be created by setting the **rhsso_external_access** parameter to **True** in the **Tackle CR** that manages the MTA instance.

For more information, see Red Hat Single Sign-On features and concepts .

3.5.1. Roles and Permissions

The following table contains the roles and permissions (scopes) that MTA seeds the managed RHSSO instance with:

tackle-admin	Resource Name	Verbs
	addons	delete get post put
	adoptionplans	post
	applications	delete get post put
	applications.facts	delete get post put
	applications.tags	delete get post put
	applications.bucket	delete get post put

	assessments	delete get patch post put
	businessservices	delete get post put
	dependencies	delete get post put
	identities	delete get post put
	imports	delete get post put
	jobfunctions	delete get post put
	proxies	delete get post put
	reviews	delete get post put
	settings	delete get post put
	stakeholdergroups	delete get post put

	stakeholders	delete get post put
	tags	delete get post put
	tagtypes	delete get post put
	tasks	delete get post put
	tasks.bucket	delete get post put
	tickets	delete get post put
	trackers	delete get post put
	cache	delete get
	files	delete get post put
	rulebundles	delete get post put
tackle-architect	**Resource Name**	**Verbs**

	addons	delete get post put
	applications.bucket	delete get post put
	adoptionplans	post
	applications	delete get post put
	applications.facts	delete get post put
	applications.tags	delete get post put
	assessments	delete get patch post put
	businessservices	delete get post put
	dependencies	delete get post put
	identities	get
	imports	delete get post put

	jobfunctions	delete get post put
	proxies	get
	reviews	delete get post put
	settings	get
	stakeholdergroups	delete get post put
	stakeholders	delete get post put
	tags	delete get post put
	tagtypes	delete get post put
	tasks	delete get post put
	tasks.bucket	delete get post put
	trackers	get
	tickets	delete get post put

	cache	get
	files	delete get post put
	rulebundles	delete get post put
tackle-migrator	**Resource Name**	**Verbs**
	addons	get
	adoptionplans	post
	applications	get
	applications.facts	get
	applications.tags	get
	applications.bucket	get
	assessments	get post
	businessservices	get
	dependencies	delete get post put
	identities	get
	imports	get
	jobfunctions	get
	proxies	get
	reviews	get post put

	settings	get
	stakeholdergroups	get
	stakeholders	get
	tags	get
	tagtypes	get
	tasks	delete get post put
	tasks.bucket	delete get post put
	tackers	get
	tickets	get
	cache	get
	files	get
	rulebundles	get

CHAPTER 4. CONFIGURING THE INSTANCE ENVIRONMENT

You can configure the following in **Administration** view:

- General
- Credentials
- Repositories
- HTTP and HTTPS proxy settings
- Custom migration targets
- Issue management

4.1. GENERAL

You can enable or disable the following options:

- Reviewing applications without running an assessment first
- Downloading HTML reports
- Downloading CSV reports

4.2. CONFIGURING CREDENTIALS

You can configure the following types of credentials in **Administration** view:

- Source control
- Maven
- Proxy

4.2.1. Configuring source control credentials

You can configure source control credentials in the **Credentials** view of the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In **Administration** view, click **Credentials**.
2. Click **Create new**.
3. Enter the following information:
    - Name
    - Description (Optional)
4. In the **Type** list, select **Source Control**.

5. In the **User credentials** list, select **Credential Type** and enter the requested information:
   - Username/Password
     - Username
     - Password (hidden)
   - SCM Private Key/Passphrase
     - SCM Private Key
     - Private Key Passphrase (hidden)

     NOTE

     Type-specific credential information such as keys and passphrases is either hidden or shown as [Encrypted].
6. Click **Create**.
   MTA validates the input and creates a new credential. SCM keys must be parsed and checked for validity. If the validation fails, the following error message is displayed: **“not a valid key/XML file”**.

4.2.2. Configuring Maven credentials

You can configure new Maven credentials in the **Credentials** view of the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In **Administration** view, click **Credentials**.
2. Click **Create new**.
3. Enter the following information:
   - Name
   - Description (Optional)
4. In the **Type** list, select **Maven Settings File.**
5. Upload the settings file or paste its contents.
6. Click **Create**.
   MTA validates the input and creates a new credential. The Maven **settings.xml** file must be parsed and checked for validity. If the validation fails, the following error message is displayed: **“not a valid key/XML file”**.

4.2.3. Configuring proxy credentials

You can configure proxy credentials in the **Credentials** view of the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In **Administration** view, click **Credentials**.
2. Click **Create new**.
3. Enter the following information:
    - Name
    - Description (Optional)
4. In the **Type** list, select **Proxy**.
5. Enter the following information.
    - Username
    - Password

    NOTE

    Type-specific credential information such as keys and passphrases is either hidden or shown as [Encrypted].

6. Click **Create**.
   MTA validates the input and creates a new credential.

4.3. CONFIGURING REPOSITORIES

You can configure the following types of repositories in **Administration** view:

- Git
- Subversion
- Maven

4.3.1. Configuring Git repositories

You can configure Git repositories in the **Repositories** view of the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In **Administration** view, click **Repositories** and then click **Git**.
2. Toggle the **Consume insecure Git repositories**switch to the right.

4.3.2. Configuring subversion repositories

You can configure subversion repositories in the **Repositories** view of the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In **Administration** view, click **Repositories** and then click **Subversion**.

2. Toggle the **Consume insecure Subversion repositories** switch to the right.

4.3.3. Configuring a Maven repository and reducing its size

You can use the MTA user interface to both configure a Maven repository and to reduce its size.

4.3.3.1. Configuring a Maven repository

You can configure a Maven repository in the **Repositories** view of the Migration Toolkit for Applications (MTA) user interface.

> NOTE
>
> If the **rwx_supported** configuration option of the Tackle CR is set to **false**, the **Consume insecure artifact repositories** switch is disabled and this procedure is not possible.

Procedure

1. In **Administration** view, click **Repositories** and then click **Maven**.

2. Toggle the **Consume insecure artifact repositories** switch to the right.

4.3.3.2. Reducing the size of a Maven repository

You can reduce the size of a Maven repository in the **Repositories** view of the Migration Toolkit for Applications (MTA) user interface.

> NOTE
>
> If the **rwx_supported** configuration option of the Tackle CR is set to **false**, both the **Local artifact repository** field and the **Clear repository** button are disabled and this procedure is not possible.

Procedure

1. In **Administration** view, click **Repositories** and then click **Maven**.

2. Click the **Clear repository** link.

   > NOTE
   >
   > Depending on the size of the repository, the size change may not be evident despite the function working properly.

4.4. CONFIGURING HTTP AND HTTPS PROXY SETTINGS

You can configure HTTP and HTTPS proxy settings with this management module.

Procedure

1. In the **Administration** view, click **Proxy**.

2. Toggle **HTTP proxy** or **HTTPS proxy** to enable the proxy connection.
3. Enter the following information:
   - Proxy host
   - Proxy port
4. Optional: Toggle **HTTP proxy credentials** or **HTTPS proxy credentials** to enable authentication.
5. Click **Insert**.

4.5. SEEDING AN INSTANCE

If you are a project architect, you can configure the instance’s key parameters in the Controls window, before migration. The parameters can be added and edited as needed. The following parameters define applications, individuals, teams, verticals or areas within an organization affected or participating in the migration:

- Stakeholders
- Stakeholder groups
- Job functions
- Business services
- Tag categories
- Tags

You can create and configure an instance in any order. However, the suggested order below is the most efficient for creating stakeholders and tags.

Stakeholders:

1. Create Stakeholder groups
2. Create Job functions
3. Create Stakeholders

Tags:

1. Create Tag categories
2. Create Tags

Stakeholders and defined by:

- Email
- Name
- Job function

- Stakeholder groups

4.5.1. Creating a new stakeholder group

There are no default stakeholder groups defined. You can create a new stakeholder group by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Stakeholder groups**.
3. Click **Create new**.
4. Enter the following information:
    - Name
    - Description
    - Member(s)
5. Click **Create**.

4.5.2. Creating a new job function

Migration Toolkit for Applications (MTA) uses the job function attribute to classify stakeholders and provides a list of default values that can be expanded.

You can create a new job function, which is not in the default list, by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Job functions**.
3. Click **Create new**.
4. Enter a job function title in the **Name** text box.
5. Click **Create**.

4.5.3. Creating a new stakeholder

You can create a new migration project stakeholder by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Stakeholders**.
3. Click **Create new**.

4. Enter the following information:
   - Email
   - Name
   - Job function - custom functions can be created
   - Stakeholder group
5. Click **Create**.

4.5.4. Creating a new business service

Migration Toolkit for Applications (MTA) uses the business service attribute to specify the departments within the organization that use the application and that are affected by the migration.

You can create a new business service by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Business services**.
3. Click **Create new**.
4. Enter the following information:
   - Name
   - Description
   - Owner
5. Click **Create**.

4.5.5. Creating new tag categories

Migration Toolkit for Applications (MTA) uses tags in multiple categories and provides a list of default values. You can create a new tag category by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Tags**.
3. Click **Create tag category**.
4. Enter the following information:
   - Name
   - Rank - the order in which the tags appear on the applications
   - Color

5. Click **Create**.

4.5.5.1. Creating new tags

You can create a new tag, which is not in the default list, by following the procedure below.

Procedure

1. In **Migration** view, click **Controls**.
2. Click **Tags**.
3. Click **Create tag**.
4. Enter the following information:
   - Name
   - Tag category
5. Click **Create**.

CHAPTER 5. CREATING AND CONFIGURING A JIRA CONNECTION

You can track application migrations by creating a Jira issue for each migration from within the MTA user interface. To be able to create Jira issues, you first need to do the following:

1. Create an MTA credential to authenticate to the API of the Jira instance that you create in the next step.

2. Create a Jira instance in MTA and establish a connection to that instance.

5.1. CONFIGURING JIRA CREDENTIALS

To define a Jira instance in MTA and establish a connection to that instance, you must first create an MTA credential to authenticate to the Jira instance’s API.

Two types of credentials are available:

- **Basic auth** - for Jira Cloud and a private Jira server or data center

- **Bearer Token** - for a private Jira server or data center

To create an MTA credential, follow the procedure below.

Procedure

1. In **Administration** view, click **Credentials**.
   The **Credentials** page opens.

2. Click **Create new**.

3. Enter the following information:

   - **Name**

   - **Description** (optional)

4. In the **Type** list, select **Basic Auth (Jira)** or **Bearer Token (Jira)**:

   - If you selected **Basic Auth (Jira)**, proceed as follows:

     a. In the **Email** field, enter your email.

     b. In the **Token** field, depending on the specific Jira configuration, enter either your token generated on the Jira site or your Jira login password.

        NOTE

        To obtain a Jira token, you need to log in to the Jira site.

     c. Click **Save**.
        The new credential appears on the **Credentials** page.

   - If you selected **Bearer Token (Jira)**, proceed as follows:

a. In the **Token** field, enter your token generated on the Jira site.

b. Click **Save**.
The new credential appears on the **Credentials** page.

You can edit a credential by clicking **Edit**.

To delete a credential, click **Delete**.

> **NOTE**
>
> You cannot delete a credential that has already been assigned to a Jira connection instance.

5.2. CREATING AND CONFIGURING A JIRA CONNECTION

To create a Jira instance in MTA and establish a connection to that instance, follow the procedure below.

Procedure

1. In **Administration** view, under **Issue Management**, click **Jira**.
   The **Jira configuration** page opens.

2. Click **Create new**.
   The **New instance** window opens.

3. Enter the following information:

   - Name of the instance
   - URL of the web interface of your Jira account
   - Instance type - select either **Jira Cloud** or **Jira Server/Data center** from the list
   - Credentials - select from the list

   > **NOTE**
   >
   > If the selected instance type is **Jira Cloud**, only **Basic Auth** credentials are displayed in the list.
   >
   > If the selected instance type is **Jira Server/Data center**, both **Basic Auth** and **Token Bearer** credentials are displayed. Choose the type that is appropriate for the particular configuration of your Jira server or data center.

4. By default, a connection cannot be established with a server with an invalid certificate. To override this restriction, toggle the **Enable insecure communication** switch.

5. Click **Create**.
   The new connection instance appears on the **Jira configuration** page.

   Once the connection has been established and authorized, the status in the **Connection** column becomes **Connected**.

If the **Connection** status becomes **Not connected**, click the status to see the reason for the error.

The **Jira configuration** table has filtering by **Name** and **URL** and sorting by **Instance name** and **URL**.

NOTE

A Jira connection that was used for creating issues for a migration wave cannot be removed as long as the issues exist in Jira, even after the migration wave is deleted.

CHAPTER 6. ASSESSING AND ANALYZING APPLICATIONS WITH MTA

You can use the Migration Toolkit for Applications (MTA) user interface to assess and analyze applications:

- When assessing applications, MTA estimates the risks and costs involved in preparing applications for containerization, including time, personnel, and other factors. You can use the results of an assessment for discussions between stakeholders to determine whether applications are suitable for containerization.
- When analyzing applications, MTA uses rules to determine which specific lines in an application must be modified before the application can be migrated or modernized.

6.1. THE ASSESSMENT MODULE FEATURES

The Migration Toolkit for Applications (MTA) **Assessment** module offers the following features for assessing and analyzing applications:

Assessment hub

The **Assessment** hub integrates with the **Application inventory**.

Enhanced assessment questionnaire capabilities

In MTA 7.0, you can import and export assessment questionnaires. You can also design custom questionnaires with a downloadable template by using the YAML syntax, which includes the following features:

- Conditional questions: You can include or exclude questions based on the application or archetype if a certain tag is present on this application or archetype.
- Application auto-tagging based on answers: You can define tags to be applied to applications or archetypes if a certain answer was provided.
- Automated answers from tags in applications or archetypes.

For more information, see The custom assessment questionnaire.

NOTE

You can customize and save the default questionnaire. For more information, see The default assessment questionnaire.

Multiple assessment questionnaires

The **Assessment** module supports multiple questionnaires, relevant to one or more applications.

Archetypes

You can group applications with similar characteristics into archetypes. This allows you to assess multiple applications at once. Each archetype has a shared taxonomy of tags, stakeholders, and stakeholder groups. All applications inherit assessment and review from their assigned archetypes. For more information, see Working with archetypes.

6.2. MTA ASSESSMENT QUESTIONNAIRES

The Migration Toolkit for Applications (MTA) uses an assessment questionnaire, either default or custom, to assess the risks involved in containerizing an application.

The assessment report provides information about applications and risks associated with migration. The report also generates an adoption plan informed by the prioritization, business criticality, and dependencies of the applications submitted for assessment.

6.2.1. The default assessment questionnaire

Legacy Pathfinder is the default Migration Toolkit for Applications (MTA) questionnaire. Pathfinder is a questionnaire-based tool that you can use to evaluate the suitability of applications for modernization in containers on an enterprise Kubernetes platform.

Through interaction with the default questionnaire and the review process, the system is enriched with application knowledge exposed through the collection of assessment reports.

You can export the default questionnaire to a YAML file:

Example 6.1. The Legacy Pathfinder YAML file

```yaml
name: Legacy Pathfinder
description: ''
sections:
  - order: 1
    name: Application details
    questions:
      - order: 1
        text: >-
          Does the application development team understand and actively develop
          the application?
        explanation: >-
          How much knowledge does the team have about the application's
          development or usage?
        answers:
          - order: 2
            text: >-
              Maintenance mode, no SME knowledge or adequate documentation
              available
            risk: red
            rationale: ''
            mitigation: ''
          - order: 0
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 1
            text: >-
              Little knowledge, no development (example: third-party or
              commercial off-the-shelf application)
            risk: red
            rationale: ''
            mitigation: ''
          - order: 3
            text: Maintenance mode, SME knowledge is available
            risk: yellow
      rationale: ''
      mitigation: ''
    - order: 4
      text: Actively developed, SME knowledge is available
      risk: green
      rationale: ''
      mitigation: ''
    - order: 5
      text: greenfield application
      risk: green
      rationale: ''
      mitigation: ''
- order: 2
  text: How is the application supported in production?
  explanation: >-
    Does the team have sufficient knowledge to support the application in
    production?
  answers:
    - order: 3
      text: >-
        Multiple teams provide support using an established escalation
        model
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: >-
        External support provider with a ticket-driven escalation process;
        no inhouse support resources
      risk: red
      rationale: ''
      mitigation: ''
    - order: 2
      text: >-
        Separate internal support team, separate from the development
        team, with little interaction between the teams
      risk: red
      rationale: ''
      mitigation: ''
    - order: 4
      text: >-
        SRE (Site Reliability Engineering) approach with a knowledgeable
        and experienced operations team
      risk: green
      rationale: ''
      mitigation: ''
    - order: 5
      text: >-
        DevOps approach with the same team building the application and
        supporting it in production
      risk: green
      rationale: ''
      mitigation: ''
- order: 3
  text: >-
    How much time passes from when code is committed until the application
    is deployed to production?
  explanation: What is the development latency?
  answers:
    - order: 3
      text: 2-6 months
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: Not tracked
      risk: red
      rationale: ''
      mitigation: ''
    - order: 2
      text: More than 6 months
      risk: red
      rationale: ''
      mitigation: ''
    - order: 4
      text: 8-30 days
      risk: green
      rationale: ''
      mitigation: ''
    - order: 5
      text: 1-7 days
      risk: green
      rationale: ''
      mitigation: ''
    - order: 6
      text: Less than 1 day
      risk: green
      rationale: ''
      mitigation: ''
- order: 4
  text: How often is the application deployed to production?
  explanation: Deployment frequency
  answers:
    - order: 3
      text: Between once a month and once every 6 months
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
```

```
          mitigation: ''
        - order: 1
          text: Not tracked
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: Less than once every 6 months
          risk: red
          rationale: ''
          mitigation: ''
        - order: 4
          text: Weekly
          risk: green
          rationale: ''
          mitigation: ''
        - order: 5
          text: Daily
          risk: green
          rationale: ''
          mitigation: ''
        - order: 6
          text: Several times a day
          risk: green
          rationale: ''
          mitigation: ''
    - order: 5
      text: >-
        What is the application's mean time to recover (MTTR) from failure in
        a production environment?
      explanation: Average time for the application to recover from failure
      answers:
        - order: 5
          text: Less than 1 hour
          risk: green
          rationale: ''
          mitigation: ''
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: Not tracked
          risk: red
          rationale: ''
          mitigation: ''
        - order: 3
          text: 1-7 days
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 2
          text: 1 month or more
          risk: red
          rationale: ''
      mitigation: ''
    - order: 4
      text: 1-24 hours
      risk: green
      rationale: ''
      mitigation: ''
- order: 6
  text: Does the application have legal and/or licensing requirements?
  explanation: >-
    Legal and licensing requirements must be assessed to determine their
    possible impact (cost, fault reporting) on the container platform
    hosting the application. Examples of legal requirements: isolated
    clusters, certifications, compliance with the Payment Card Industry
    Data Security Standard or the Health Insurance Portability and
    Accountability Act. Examples of licensing requirements: per server,
    per CPU.
  answers:
    - order: 1
      text: Multiple legal and licensing requirements
      risk: red
      rationale: ''
      mitigation: ''
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 2
      text: 'Licensing requirements (examples: per server, per CPU)'
      risk: red
      rationale: ''
      mitigation: ''
    - order: 3
      text: >-
        Legal requirements (examples: cluster isolation, hardware, PCI or
        HIPAA compliance)
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 4
      text: None
      risk: green
      rationale: ''
      mitigation: ''
- order: 7
  text: Which model best describes the application architecture?
  explanation: Describe the application architecture in simple terms.
  answers:
    - order: 3
      text: >-
        Complex monolith, strict runtime dependency startup order,
        non-resilient architecture
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 0
```

```
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 5
            text: Independently deployable components
            risk: green
            rationale: ''
            mitigation: ''
          - order: 1
            text: >-
              Massive monolith (high memory and CPU usage), singleton
              deployment, vertical scale only
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 2
            text: >-
              Massive monolith (high memory and CPU usage), non-singleton
              deployment, complex to scale horizontally
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 4
            text: 'Resilient monolith (examples: retries, circuit breakers)'
            risk: green
            rationale: ''
            mitigation: ''
  - order: 2
    name: Application dependencies
    questions:
      - order: 1
        text: Does the application require specific hardware?
        explanation: >-
          OpenShift Container Platform runs only on x86, IBM Power, or IBM Z
          systems
        answers:
          - order: 3
            text: 'Requires specific computer hardware (examples: GPUs, RAM, HDDs)'
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 0
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 1
            text: Requires CPU that is not supported by red Hat
            risk: red
            rationale: ''
            mitigation: ''
          - order: 2
            text: 'Requires custom or legacy hardware (example: USB device)'
            risk: red
            rationale: ''
          mitigation: ''
        - order: 4
          text: Requires CPU that is supported by red Hat
          risk: green
          rationale: ''
          mitigation: ''
    - order: 2
      text: What operating system does the application require?
      explanation: >-
        Only Linux and certain Microsoft Windows versions are supported in
        containers. Check the latest versions and requirements.
      answers:
        - order: 4
          text: Microsoft Windows
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: >-
            Operating system that is not compatible with OpenShift Container
            Platform (examples: OS X, AIX, Unix, Solaris)
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: Linux with custom kernel drivers or a specific kernel version
          risk: red
          rationale: ''
          mitigation: ''
        - order: 3
          text: 'Linux with custom capabilities (examples: seccomp, root access)'
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 5
          text: Standard Linux distribution
          risk: green
          rationale: ''
          mitigation: ''
    - order: 3
      text: >-
        Does the vendor provide support for a third-party component running in
        a container?
      explanation: Will the vendor support a component if you run it in a container?
      answers:
        - order: 2
          text: No vendor support for containers
          risk: red
          rationale: ''
          mitigation: ''
        - order: 0
```

```
          text: unknown
          risk: unknown
          rationale: "
          mitigation: "
        - order: 1
          text: Not recommended to run the component in a container
          risk: red
          rationale: "
          mitigation: "
        - order: 3
          text: >-
            Vendor supports containers but with limitations (examples:
            functionality is restricted, component has not been tested)
          risk: yellow
          rationale: "
          mitigation: "
        - order: 4
          text: >-
            Vendor supports their application running in containers but you
            must build your own images
          risk: yellow
          rationale: "
          mitigation: "
        - order: 5
          text: Vendor fully supports containers, provides certified images
          risk: green
          rationale: "
          mitigation: "
        - order: 6
          text: No third-party components required
          risk: green
          rationale: "
          mitigation: "
    - order: 4
      text: Incoming/northbound dependencies
      explanation: Systems or applications that call the application
      answers:
        - order: 3
          text: >-
            Many dependencies exist, can be changed because the systems are
            internally managed
          risk: green
          rationale: "
          mitigation: "
        - order: 0
          text: unknown
          risk: unknown
          rationale: "
          mitigation: "
        - order: 4
          text: Internal dependencies only
          risk: green
          rationale: "
          mitigation: "
        - order: 1
          text: >-
```

```
            Dependencies are difficult or expensive to change because they are
            legacy or third-party
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: >-
            Many dependencies exist, can be changed but the process is
            expensive and time-consuming
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 5
          text: No incoming/northbound dependencies
          risk: green
          rationale: ''
          mitigation: ''
    - order: 5
      text: Outgoing/southbound dependencies
      explanation: Systems or applications that the application calls
      answers:
        - order: 3
          text: Application not ready until dependencies are verified available
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: >-
            Dependency availability only verified when application is
            processing traffic
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: Dependencies require a complex and strict startup order
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 4
          text: Limited processing available if dependencies are unavailable
          risk: green
          rationale: ''
          mitigation: ''
        - order: 5
          text: No outgoing/southbound dependencies
          risk: green
          rationale: ''
          mitigation: ''
- order: 3
  name: Application architecture
  questions:
```

```
- order: 1
  text: >-
    How resilient is the application? How well does it recover from
    outages and restarts?
  explanation: >-
    If the application or one of its dependencies fails, how does the
    application recover from failure? Is manual intervention required?
  answers:
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: >-
        Application cannot be restarted cleanly after failure, requires
        manual intervention
      risk: red
      rationale: ''
      mitigation: ''
    - order: 2
      text: >-
        Application fails when a soutbound dependency is unavailable and
        does not recover automatically
      risk: red
      rationale: ''
      mitigation: ''
    - order: 3
      text: >-
        Application functionality is limited when a dependency is
        unavailable but recovers when the dependency is available
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 4
      text: >-
        Application employs resilient architecture patterns (examples:
        circuit breakers, retry mechanisms)
      risk: green
      rationale: ''
      mitigation: ''
    - order: 5
      text: >-
        Application containers are randomly terminated to test resiliency;
        chaos engineering principles are followed
      risk: green
      rationale: ''
      mitigation: ''
- order: 2
  text: How does the external world communicate with the application?
  explanation: >-
    What protocols do external clients use to communicate with the
    application?
  answers:
    - order: 0
      text: unknown
```

```
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 1
            text: 'Non-TCP/IP protocols (examples: serial, IPX, AppleTalk)'
            risk: red
            rationale: ''
            mitigation: ''
          - order: 2
            text: TCP/IP, with host name or IP address encapsulated in the payload
            risk: red
            rationale: ''
            mitigation: ''
          - order: 3
            text: 'TCP/UDP without host addressing (example: SSH)'
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 4
            text: TCP/UDP encapsulated, using TLS with SNI header
            risk: green
            rationale: ''
            mitigation: ''
          - order: 5
            text: HTTP/HTTPS
            risk: green
            rationale: ''
            mitigation: ''
      - order: 3
        text: How does the application manage its internal state?
        explanation: >-
          If the application must manage or retain an internal state, how is
          this done?
        answers:
          - order: 0
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 3
            text: State maintained in non-shared, non-ephemeral storage
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 1
            text: Application components use shared memory within a pod
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 2
            text: >-
              State is managed externally by another product (examples:
              Zookeeper or red Hat Data Grid)
            risk: yellow
            rationale: ''
            mitigation: ''
```

```
  - order: 4
    text: Disk shared between application instances
    risk: green
    rationale: ''
    mitigation: ''
  - order: 5
    text: Stateless or ephemeral container storage
    risk: green
    rationale: ''
    mitigation: ''
- order: 4
  text: How does the application handle service discovery?
  explanation: How does the application discover services?
  answers:
  - order: 0
    text: unknown
    risk: unknown
    rationale: ''
    mitigation: ''
  - order: 1
    text: >-
      Uses technologies that are not compatible with Kubernetes
      (examples: hardcoded IP addresses, custom cluster manager)
    risk: red
    rationale: ''
    mitigation: ''
  - order: 2
    text: >-
      Requires an application or cluster restart to discover new service
      instances
    risk: red
    rationale: ''
    mitigation: ''
  - order: 3
    text: >-
      Uses technologies that are compatible with Kubernetes but require
      specific libraries or services (examples: HashiCorp Consul,
      Netflix Eureka)
    risk: yellow
    rationale: ''
    mitigation: ''
  - order: 4
    text: Uses Kubernetes DNS name resolution
    risk: green
    rationale: ''
    mitigation: ''
  - order: 5
    text: Does not require service discovery
    risk: green
    rationale: ''
    mitigation: ''
- order: 5
  text: How is the application clustering managed?
  explanation: >-
    Does the application require clusters? If so, how is clustering
    managed?
```

```
      answers:
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: 'Manually configured clustering (example: static clusters)'
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: Managed by an external off-PaaS cluster manager
          risk: red
          rationale: ''
          mitigation: ''
        - order: 3
          text: >-
            Managed by an application runtime that is compatible with
            Kubernetes
          risk: green
          rationale: ''
          mitigation: ''
        - order: 4
          text: No cluster management required
          risk: green
          rationale: ''
          mitigation: ''
- order: 4
  name: Application observability
  questions:
    - order: 1
      text: How does the application use logging and how are the logs accessed?
      explanation: How the application logs are accessed
      answers:
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: Logs are unavailable or are internal with no way to export them
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: >-
            Logs are in a custom binary format, exposed with non-standard
            protocols
          risk: red
          rationale: ''
          mitigation: ''
        - order: 3
          text: Logs are exposed using syslog
          risk: yellow
          rationale: ''
      mitigation: ''
    - order: 4
      text: Logs are written to a file system, sometimes as multiple files
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 5
      text: 'Logs are forwarded to an external logging system (example: Splunk)'
      risk: green
      rationale: ''
      mitigation: ''
    - order: 6
      text: 'Logs are configurable (example: can be sent to stdout)'
      risk: green
      rationale: ''
      mitigation: ''
- order: 2
  text: Does the application provide metrics?
  explanation: >-
    Are application metrics available, if necessary (example: OpenShift
    Container Platform collects CPU and memory metrics)?
  answers:
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: No metrics available
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 2
      text: Metrics collected but not exposed externally
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 3
      text: 'Metrics exposed using binary protocols (examples: SNMP, JMX)'
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 4
      text: >-
        Metrics exposed using a third-party solution (examples: Dynatrace,
        AppDynamics)
      risk: green
      rationale: ''
      mitigation: ''
    - order: 5
      text: >-
        Metrics collected and exposed with built-in Prometheus endpoint
        support
      risk: green
      rationale: ''
      mitigation: ''
```

```
- order: 3
  text: >-
    How easy is it to determine the application's health and readiness to
    handle traffic?
  explanation: >-
    How do we determine an application's health (liveness) and readiness
    to handle traffic?
  answers:
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: No health or readiness query functionality available
      risk: red
      rationale: ''
      mitigation: ''
    - order: 3
      text: Basic application health requires semi-complex scripting
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 4
      text: Dedicated, independent liveness and readiness endpoints
      risk: green
      rationale: ''
      mitigation: ''
    - order: 2
      text: Monitored and managed by a custom watchdog process
      risk: red
      rationale: ''
      mitigation: ''
    - order: 5
      text: Health is verified by probes running synthetic transactions
      risk: green
      rationale: ''
      mitigation: ''
- order: 4
  text: What best describes the application's runtime characteristics?
  explanation: >-
    How would the profile of an application appear during runtime
    (examples: graphs showing CPU and memory usage, traffic patterns,
    latency)? What are the implications for a serverless application?
  answers:
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: >-
        Deterministic and predictable real-time execution or control
        requirements
      risk: red
      rationale: ''
    mitigation: ''
  - order: 2
    text: >-
      Sensitive to latency (examples: voice applications, high frequency
      trading applications)
    risk: yellow
    rationale: ''
    mitigation: ''
  - order: 3
    text: Constant traffic with a broad range of CPU and memory usage
    risk: yellow
    rationale: ''
    mitigation: ''
  - order: 4
    text: Intermittent traffic with predictable CPU and memory usage
    risk: green
    rationale: ''
    mitigation: ''
  - order: 5
    text: Constant traffic with predictable CPU and memory usage
    risk: green
    rationale: ''
    mitigation: ''
- order: 5
  text: How long does it take the application to be ready to handle traffic?
  explanation: How long the application takes to boot
  answers:
  - order: 0
    text: unknown
    risk: unknown
    rationale: ''
    mitigation: ''
  - order: 1
    text: More than 5 minutes
    risk: red
    rationale: ''
    mitigation: ''
  - order: 2
    text: 2-5 minutes
    risk: yellow
    rationale: ''
    mitigation: ''
  - order: 3
    text: 1-2 minutes
    risk: yellow
    rationale: ''
    mitigation: ''
  - order: 4
    text: 10-60 seconds
    risk: green
    rationale: ''
    mitigation: ''
  - order: 5
    text: Less than 10 seconds
    risk: green
    rationale: ''
```

```
          mitigation: ''
- order: 5
  name: Application cross-cutting concerns
  questions:
    - order: 1
      text: How is the application tested?
      explanation: >-
        Is the application is tested? Is it easy to test (example: automated
        testing)? Is it tested in production?
      answers:
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: No testing or minimal manual testing only
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: Minimal automated testing, focused on the user interface
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 3
          text: >-
            Some automated unit and regression testing, basic CI/CD pipeline
            testing; modern test practices are not followed
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 4
          text: >-
            Highly repeatable automated testing (examples: unit, integration,
            smoke tests) before deploying to production; modern test practices
            are followed
          risk: green
          rationale: ''
          mitigation: ''
        - order: 5
          text: >-
            Chaos engineering approach, constant testing in production
            (example: A/B testing + experimentation)
          risk: green
          rationale: ''
          mitigation: ''
    - order: 2
      text: How is the application configured?
      explanation: >-
        How is the application configured? Is the configuration method
        appropriate for a container? External servers are runtime
        dependencies.
      answers:
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 1
          text: >-
            Configuration files compiled during installation and configured
            using a user interface
          risk: red
          rationale: ''
          mitigation: ''
        - order: 2
          text: >-
            Configuration files are stored externally (example: in a database)
            and accessed using specific environment keys (examples: host name,
            IP address)
          risk: red
          rationale: ''
          mitigation: ''
        - order: 3
          text: Multiple configuration files in multiple file system locations
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 4
          text: >-
            Configuration files built into the application and enabled using
            system properties at runtime
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 5
          text: >-
            Configuration retrieved from an external server (examples: Spring
            Cloud Config Server, HashiCorp Consul)
          risk: yellow
          rationale: ''
          mitigation: ''
        - order: 6
          text: >-
            Configuration loaded from files in a single configurable location;
            environment variables used
          risk: green
          rationale: ''
          mitigation: ''
    - order: 4
      text: How is the application deployed?
      explanation: >-
        How the application is deployed and whether the deployment process is
        suitable for a container platform
      answers:
        - order: 0
          text: unknown
          risk: unknown
          rationale: ''
          mitigation: ''
        - order: 3
      text: Simple automated deployment scripts
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 1
      text: Manual deployment using a user interface
      risk: red
      rationale: ''
      mitigation: ''
    - order: 2
      text: Manual deployment with some automation
      risk: red
      rationale: ''
      mitigation: ''
    - order: 4
      text: >-
        Automated deployment with manual intervention or complex promotion
        through pipeline stages
      risk: yellow
      rationale: ''
      mitigation: ''
    - order: 5
      text: >-
        Automated deployment with a full CI/CD pipeline, minimal
        intervention for promotion through pipeline stages
      risk: green
      rationale: ''
      mitigation: ''
    - order: 6
      text: Fully automated (GitOps), blue-green, or canary deployment
      risk: green
      rationale: ''
      mitigation: ''
- order: 5
  text: Where is the application deployed?
  explanation: Where does the application run?
  answers:
    - order: 0
      text: unknown
      risk: unknown
      rationale: ''
      mitigation: ''
    - order: 1
      text: Bare metal server
      risk: green
      rationale: ''
      mitigation: ''
    - order: 2
      text: 'Virtual machine (examples: red Hat Virtualization, VMware)'
      risk: green
      rationale: ''
      mitigation: ''
    - order: 3
      text: 'Private cloud (example: red Hat OpenStack Platform)'
      risk: green
      rationale: ''
```

```
            mitigation: ''
          - order: 4
            text: >-
              Public cloud provider (examples: Amazon Web Services, Microsoft
              Azure, Google Cloud Platform)
            risk: green
            rationale: ''
            mitigation: ''
          - order: 5
            text: >-
              Platform as a service (examples: Heroku, Force.com, Google App
              Engine)
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 7
            text: Other. Specify in the comments field
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 6
            text: Hybrid cloud (public and private cloud providers)
            risk: green
            rationale: ''
            mitigation: ''
      - order: 6
        text: How mature is the containerization process, if any?
        explanation: If the team has used containers in the past, how was it done?
        answers:
          - order: 0
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 1
            text: Application runs in a container on a laptop or desktop
            risk: red
            rationale: ''
            mitigation: ''
          - order: 3
            text: Some experience with containers but not yet fully defined
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 4
            text: >-
              Proficient with containers and container platforms (examples:
              Swarm, Kubernetes)
            risk: green
            rationale: ''
            mitigation: ''
          - order: 5
            text: Application containerization has not yet been attempted
            risk: green
            rationale: ''
            mitigation: ''
      - order: 3
        text: How does the application acquire security keys or certificates?
        explanation: >-
          How does the application retrieve credentials, keys, or certificates?
          External systems are runtime dependencies.
        answers:
          - order: 0
            text: unknown
            risk: unknown
            rationale: ''
            mitigation: ''
          - order: 1
            text: Hardware security modules or encryption devices
            risk: red
            rationale: ''
            mitigation: ''
          - order: 2
            text: >-
              Keys/certificates bound to IP addresses and generated at runtime
              for each application instance
            risk: red
            rationale: ''
            mitigation: ''
          - order: 3
            text: Keys/certificates compiled into the application
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 4
            text: Loaded from a shared disk
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 5
            text: >-
              Retrieved from an external server (examples: HashiCorp Vault,
              CyberArk Conjur)
            risk: yellow
            rationale: ''
            mitigation: ''
          - order: 6
            text: Loaded from files
            risk: green
            rationale: ''
            mitigation: ''
          - order: 7
            text: Not required
            risk: green
            rationale: ''
            mitigation: ''
thresholds:
  red: 5
  yellow: 30
  unknown: 5
riskMessages:
  red: ''
  yellow: ''
  green: ''
  unknown: ''
builtin: true
```

6.2.2. The custom assessment questionnaire

You can use the Migration Toolkit for Applications (MTA) to import a custom assessment questionnaire by using a custom YAML syntax to define the questionnaire. The YAML syntax supports the following features:

Conditional questions

The YAML syntax supports including or excluding questions based on tags existing on the application or archetype. For example, if the application or archetype has the **Language/Java** tag, the **What is the main JAVA framework used in your application?** question is included in the questionnaire:

```
...
 questions:
  - order: 1
   text: What is the main JAVA framework used in your application?
   explanation: Identify the primary JAVA framework used in your application.
   includeFor:
    - category: Language
     tag: Java
...
```

Automated answers based on tags present on the assessed application or archetype

Automated answers are selected based on the tags existing on the application or archetype. For example, if an application or archetype has the **Runtime/Quarkus** tag, the **Quarkus** answer is automatically selected, and if an application or archetype has the **Runtime/Spring Boot** tag, the **Spring Boot** answer is automatically selected:

```
...
 text: What is the main technology in your application?
  explanation: Identify the main framework or technology used in your application.
   answers:
    - order: 1
     text: Quarkus
     risk: green
     autoAnswerFor:
      - category: Runtime
       tag: Quarkus
    - order: 2
     text: Spring Boot
     risk: green
     autoAnswerFor:
      - category: Runtime
       tag: Spring Boot
...
```

Autotagging of applications based on answers

With this feature, tags are automatically applied to the assessed application or archetype based on the answer if this answer is selected. The tags are transitive. Each tag is defined by the following elements:

- **category**: Category of the target tag (**String**).
- **tag**: Definition for the target tag as (**String**).

For example, if the selected answer is **Quarkus**, the **Runtime/Quarkus** tag is applied to the assessed application or archetype. If the selected answer is **Spring Boot**, the **Runtime/Spring Boot** tag is applied to the assessed application or archetype:

```
...
questions:
  - order: 1
    text: What is the main technology in your application?
    explanation: Identify the main framework or technology used in your application.
    answers:
      - order: 1
        text: Quarkus
        risk: green
        applyTags:
          - category: Runtime
            tag: Quarkus
      - order: 2
        text: Spring Boot
        risk: green
        applyTags:
          - category: Runtime
            tag: Spring Boot
...
```

6.2.2.1. The YAML template for the custom questionnaire

You can use the following YAML template to build your custom questionnaire. You can download this template by clicking **Download YAML template** on the **Assessment questionnaires** page.

Example 6.2. The YAML template for the custom questionnaire

```
name: Uploadable Cloud Readiness Questionnaire Template
description: This questionnaire is an example template for assessing cloud readiness. It serves as
a guide for users to create and customize their own questionnaire templates.
required: true
sections:
  - order: 1
    name: Application Technologies
    questions:
      - order: 1
        text: What is the main technology in your application?
        explanation: Identify the main framework or technology used in your application.
        includeFor:
          - category: Language
            tag: Java
        answers:
          - order: 1
        text: Quarkus
        risk: green
        rationale: Quarkus is a modern, container-friendly framework.
        mitigation: No mitigation needed.
        applyTags:
          - category: Runtime
            tag: Quarkus
        autoAnswerFor:
          - category: Runtime
            tag: Quarkus
      - order: 2
        text: Spring Boot
        risk: green
        rationale: Spring Boot is versatile and widely used.
        mitigation: Ensure container compatibility.
        applyTags:
          - category: Runtime
            tag: Spring Boot
        autoAnswerFor:
          - category: Runtime
            tag: Spring Boot
      - order: 3
        text: Legacy Monolithic Application
        risk: red
        rationale: Legacy monoliths are challenging for cloud adaptation.
        mitigation: Consider refactoring into microservices.
  - order: 2
    text: Does your application use a microservices architecture?
    explanation: Assess if the application is built using a microservices architecture.
    answers:
      - order: 1
        text: Yes
        risk: green
        rationale: Microservices are well-suited for cloud environments.
        mitigation: Continue monitoring service dependencies.
      - order: 2
        text: No
        risk: yellow
        rationale: Non-microservices architectures may face scalability issues.
        mitigation: Assess the feasibility of transitioning to microservices.
      - order: 3
        text: Unknown
        risk: unknown
        rationale: Lack of clarity on architecture can lead to unplanned issues.
        mitigation: Conduct an architectural review.

  - order: 3
    text: Is your application's data storage cloud-optimized?
    explanation: Evaluate if the data storage solution is optimized for cloud usage.
    includeFor:
      - category: Language
        tag: Java
    answers:
      - order: 1
        text: Cloud-Native Storage Solution
        risk: green
```

```
            rationale: Cloud-native solutions offer scalability and resilience.
            mitigation: Ensure regular backups and disaster recovery plans.
          - order: 2
            text: Traditional On-Premises Storage
            risk: red
            rationale: Traditional storage might not scale well in the cloud.
            mitigation: Explore cloud-based storage solutions.
          - order: 3
            text: Hybrid Storage Approach
            risk: yellow
            rationale: Hybrid solutions may have integration complexities.
            mitigation: Evaluate and optimize cloud integration points.
thresholds:
  red: 1
  yellow: 30
  unknown: 15
riskMessages:
  red: Requires deep changes in architecture or lifecycle
  yellow: Cloud friendly but needs minor changes
  green: Cloud Native
  unknown: More information needed
```

Additional resources

- The custom questionnaire fields

6.2.2.2. The custom questionnaire fields

Every custom questionnaire field marked as **required** is mandatory and must be completed. Otherwise, the YAML syntax will not validate on upload. Each subsection of the field defines a new structure or object in YAML, for example:

```
...
name: Testing
thresholds:
    red: 30
    yellow: 45
    unknown: 5
...
```

Table 6.1. The custom questionnaire fields

Questionnaire field	Description
name (required)	The name of the questionnaire. This field must be unique for the entire MTA instance.
description (optional)	A short description of the questionnaire.

Questionnaire field	Description
thresholds (required)	The definition of a threshold for each risk category of the application or archetype that is considered to be affected by that risk level. The threshold values can be the following: ● **red** (required): Numeric percentage, for example, **30** for **30%**, of red answers that the questionnaire can have until the risk level is considered red. ● **yellow** (required): Numeric percentage, for example, **30** for **30%**, of yellow answers that the questionnaire can have until the risk level is considered yellow. ● **unknown** (required): Numeric percentage, for example, **30** for **30%**, of unknown answers that the questionnaire can have until the risk level is considered unknown. The higher risk level always takes precedence. For example, if the **yellow** threshold is set to 30% and **red** to 5%, and the answers for the application or archetype are set to have 35% **yellow** and 6% **red**, the risk level for the application or archetype is red.
riskMessages (required)	Messages to be displayed in reports for each risk category. The *risk_messages* map is defined by the following fields: ● **red** (required, string): A message to be displayed in reports for the red risk level. ● **yellow** (required, string): A message to be displayed in reports for the yellow risk level. ● **green** (required, string): A message to be displayed in reports for the green risk level. ● **unknown** (required, string): A message to be displayed in reports for the unknown risk level.
sections (required)	A list of sections that the questionnaire must include. ● **name** (required, string): A name to be displayed for the section. ● **order** (required, integer): An order of the question in the section. ● **comment** (optional, string): A descripton the section. ● **questions** (required): A list of questions that belong to the section.

Questionnaire field	Description
	○ **order** (required, integer): An order of the question in the section. ○ **text** (required, string): A question to be asked. ○ **explanation** (optional, string): An additional explanation for the question. ○ **includeFor** (optional): A list that defines if a question must be displayed if any of the tags included in this list is present in the target application or archetype. ■ **category** (required, string): A category of the target tag. ■ **tag** (required, string): A target tag. ○ **excludeFor** (optional): A list that defines if a question must be skipped if any of the tags included in the list is present in the target application or archetype. ■ **category** (required, string): A category of the target tag. ■ **tag** (required, string): A target tag. ○ **answers** (required): A list of answers for the given question. ■ **order** (required, integer): An order of the question in the section. ■ **text** (required, string): An answer for the question. ■ **risk** (required): An implied risk level (red, yellow, green, or unknown) of the current answer. ■ **rationale** (optional, string): A justification for the answer that is being considered a risk. ■ **mitigation** (optional, string): An explanation of the potential mitigation strategy for the risk implied by the answer. ■ **applyTags** (optional): A list of tags to be automatically applied to the assessed application or archetype if this answer is selected. ● **category** (required, string): A category of the target tag. ● **tag** (required,string): A target tag. ■ **autoAnswerFor** (optional, list): A list of tags that will lead to this answer being automatically

Questionnaire field	Description
	selected when the application or archetype is assessed. ● **category** (required, string): A category of the target tag. ● **tag** (required, string): A target tag.

Additional resources

- The YAML template for the custom questionnaire

6.2.3. Listing available assessment questionnaires

You can use the Migration Toolkit for Applications (MTA) **Assessment** module to list the available assessment questionnaires.

Procedure

1. Select the **Administration** profile.
2. Select the **Assessment questionnaires** menu option.

Additional resources

- The default assessment questionnaire
- The custom assessment questionnaire
- Importing an assessment questionnaire
- Exporting an assessment questionnaire

6.2.4. Viewing an assessment questionnaire

You can display the questions contained in the assessment questionnaire together with the answer choices and their associated risk weight.

Procedure

1. In the **Administration** view, select **Assessment questionnaires**.
2. Click the Options menu (⋮).
3. Select **View** for the questionnaire you want to display.
4. Optional: Click the arrow to the left from the question to display the answer choices and their risk weight.

6.2.5. Exporting an assessment questionnaire

In the Migration Toolkit for Applications (MTA) 7.0, you can export an assessment questionnaire to the desired location on your system.

Procedure

1. In the **Administration** view, select **Assessment questionnaires**.
2. Select the desired questionnaire.
3. Click the Options menu (⋮).
4. Select **Export**.
5. Select the location of the download.
6. Click **Save**.

6.2.6. Importing an assessment questionnaire

In the Migration Toolkit for Applications (MTA) 7.0, you can upload an assessment questionnaire from your system.

WARNING

The name of the imported questionnaire must be unique. If the name, which is defined in the YAML syntax (**name:<name of questionnaire>**), is duplicated, the import will fail with the following error message: **UNIQUE constraint failed: Questionnaire.Name**.

Procedure

1. In the **Administration** view, select **Assessment questionnaires**.
2. Click **Import questionnaire**.
3. Click **Upload**.
4. Navigate to the location of your questionnaire.
5. Click **Open**.
6. Import the desired questionnaire by clicking **Import**.

6.2.7. Deleting an assessment questionnaire

When you delete an assessment questionnaire, its answers for all applications that use it in all archetypes are also deleted.

IMPORTANT

You cannot delete the **Legacy Pathfinder** default questionnaire.

Procedure

1. In the **Administration** view, select **Assessment questionnaires**.
2. Select the questionnaire you want to delete.
3. Click the Options menu (⋮).
4. Select **Delete**.
5. Confirm deleting by clicking on the **Name** of the questionnaire.

6.3. ASSESSING AN APPLICATION

The assessment process for applications that are to be migrated in Migration Toolkit for Applications (MTA) has been separated from other features. In the MTA user interface, you can assess an application and display the currently saved assessments by using the Assessment module.

The MTA assesses applications according to a set of questions relevant to the application, such as dependencies.

6.3.1. Creating an application by using MTA

You can create a new application by using the Migration Toolkit for Applications MTA user interface.

Prerequisites

- You are logged in to an MTA server.

Procedure

1. In the MTA user interface, select the **Migration** working mode.
2. Click **Application Inventory** in the left menu bar.
3. Click **Create new**. The **New application** dialog appears.
4. In the form, enter the following information:
   a. **Name**: A unique name for the new application.
   b. **Description**: A short description of the application (optional).
   c. **Business service**: A purpose of the application (optional).
   d. **Tags**: Software tags that characterize the application.
   e. **Owner**: A registered software owner from the drop-down list (optional).
   f. **Contributors**: Contributors from the drop-down list (optional).
   g. **Comments**: Relevant comments on the migrated application.
5. Click **Source code** and enter the following fields:
   a. **Repository type**: **Git** or **Subversion**.
   b. **Source repository**: A URL of the repository where the software code is saved.

   c. **Branch**: An application code branch in the repository (optional).

   d. **Root path**: A root path inside the repository for the target application (optional).

   NOTE: If you enter any value in either the **Branch** or **Root path** fields, the **Source repository** field becomes mandatory.

6. Optional: Click **Binary** and enter the following fields:

   a. **Group**: The Maven group for the application artifact.

   b. **Artifact**: The Maven artifact for the application.

   c. **Version**: A software version of the application.

   d. **Packaging**: The packaging for the application artifact, for example, **JAR**, **WAR**, or **EAR**.

   NOTE: If you enter any value in any of the Binary section fields, all fields automatically become mandatory.

7. Click **Create**. The dialog closes and the new application appears in the list of defined applications.

6.3.2. Opening an existing application in MTA

You can open and edit an application that has already been defined in MTA.

Prerequisites

- You are logged in to an MTA server.

Procedure

1. In the MTA user interface, select the **Migration** working mode.
2. Click **Application Inventory** in the left menu bar. A list of the available applications appears in the main pane.
3. Click **edit** (✎) to open and review the application settings.
   For the more information about the list of application settings, see Creating an application by using MTA.

6.3.3. Assessing an application by using the MTA assessment questionnaire

You can use the default Legacy Pathfinder Migration Toolkit for Applications (MTA) questionnaire to determine the risks involved in containerizing an application before reviewing the application. For more information, see The default assessment questionnaire.

IMPORTANT

You can assess only one application at a time.

Prerequisites

- You are logged in to an MTA server.

Procedure

1. In the MTA user interface, select the **Migration** view.
2. Click **Application inventory** in the left menu bar. A list of the available applications appears in the main pane.
3. Select the application you want to assess.
4. Click the **Options** menu (⋮) at the right end of the row and select **Assess** from the drop-down menu.
5. From the list of available questionnaires, click **Take** for the desired questionnaire.
6. Select **Stakeholders** and **Stakeholder groups** from the lists to track who contributed to the assessment for future reference.

   NOTE

   You can also add **Stakeholder Groups** or **Stakeholders** in the **Controls** pane of the **Migration** view. For more information, see Seeding an instance.

7. Click **Next**.
8. Answer each **Application assessment** question and click **Next**.
9. Click **Save** to review the assessment and proceed with the steps in Reviewing an application.

6.4. REVIEWING AN APPLICATION

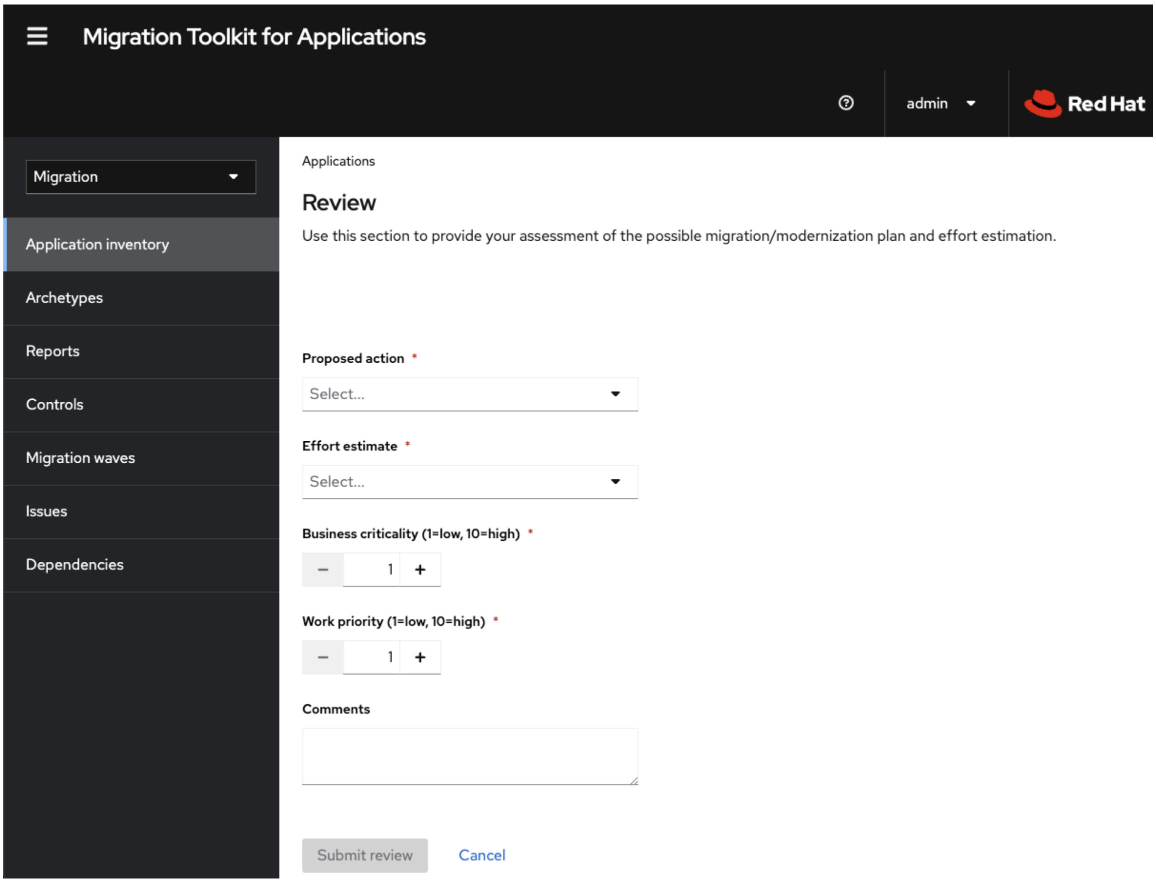
You can use the Migration Toolkit for Applications (MTA) user interface to determine the migration strategy and work priority for each application.

IMPORTANT

You can review only one application at a time.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Select the application you want to review.
3. Review the application by performing either of the following actions:
   - Click **Save and Review** while assessing the application. For more information, see Assessing an application by using the MTA assessment questionnaire.
   - Click the Options menu (⋮) at the right end of the row and select **Review** from the drop-down list. The application **Review** parameters appear in the main pane.



Review

Use this section to provide your assessment of the possible migration/modernization plan and effort estimation.

Proposed action * Select...

Effort estimate * Select...

Business criticality (1=low, 10=high) * 1

Work priority (1=low, 10=high) * 1

Comments

4. Click **Proposed action** and select the action.

5. Click **Effort estimate** and set the level of effort required to perform the assessment with the selected questionnaire.

6. In the **Business criticality** field, enter how critical the application is to the business.

7. In the **Work priority** field, enter the application's priority.

8. Optional: Enter the assessment questionnaire comments in the **Comments** field.

9. Click **Submit review**.
   The fields from **Review** are now populated on the **Application details** page.

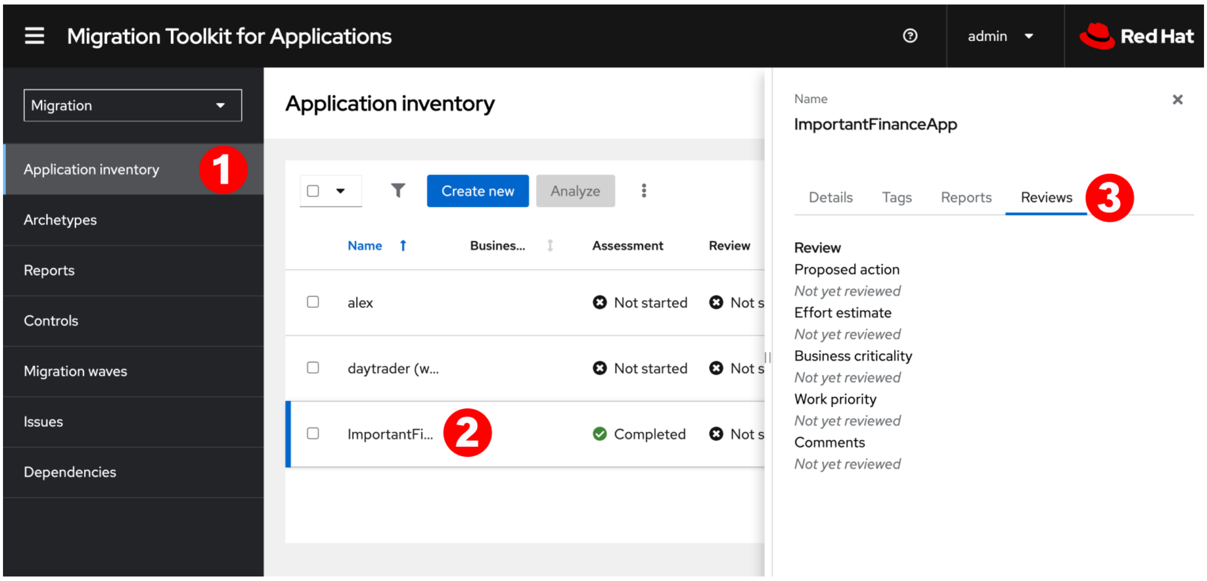
6.4.1. Displaying the review details of an application

You can display the review details of any application on the **Application inventory** page.

Procedure

1. In the **Migration** view, click **Application inventory**.

2. Select an application by opening the **Details** pane at the right of the window.



	Name	Busines...	Assessment	Review
□	alex		Not started	Not s
□	daytrader (w...		Not started	Not s
□	ImportantFi...		Completed	Not s

Name
ImportantFinanceApp

Review
Proposed action
Not yet reviewed
Effort estimate
Not yet reviewed
Business criticality
Not yet reviewed
Work priority
Not yet reviewed
Comments
Not yet reviewed

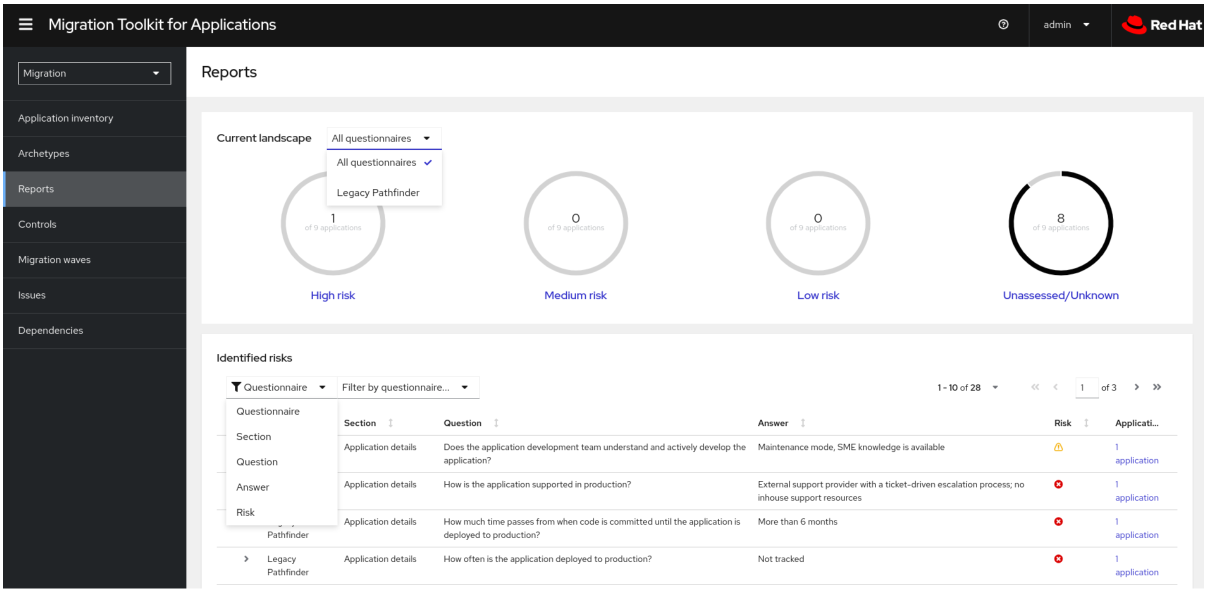
3. Click the **Reviews** tab in the pane. The application information appears, displaying the application’s questions.

6.4.2. Reviewing an assessment report

An MTA assessment report displays an aggregated assessement of the data obtained from multiple questionnaires for multiple applications.

Procedure

1. In the **Migration** view, click **Reports**. The aggregated assessment report for all applications is displayed.



	Section	Question	Answer	Risk	Applicati...
	Application details	Does the application development team understand and actively develop the application?	Maintenance mode, SME knowledge is available	⚠	1 application
	Application details	How is the application supported in production?	External support provider with a ticket-driven escalation process; no inhouse support resources	✖	1 application
Pathfinder	Application details	How much time passes from when code is committed until the application is deployed to production?	More than 6 months	✖	1 application
› Legacy Pathfinder	Application details	How often is the application deployed to production?	Not tracked	✖	1 application

2. Depending on your scenario, perform one of the following actions:

   - Display a report on the data from a particular questionnaire:

     a. Select the required questionnaire from a drop-down list of all questionnaires in the **Current landscape** pane of the report. By default, all questionnaires are selected.

     b. In the **Identified risks** pane of the report, sort the displayed list by application name, level of risk, questionnaire, questionnaire section, question, and answer.

- Display a report for a specific application:
  a. Click the link in the **Applications** column in the **Identified risks** pane of the report. The **Application inventory** page opens. The applications included in the link are displayed as a list.
  b. Click the required application. The **Assessment** side pane opens.
     - To see the assessed risk level for the application, open the **Details** tab.
     - To see the details of the assessment, open the **Reviews** tab.

6.5. TAGGING AN APPLICATION

You can attach various tags to the application that you want to analyze. The tags allow classifying applications in multiple dimensions. Tagging can be either manual or automatic.

6.5.1. Manually tagging an application

You can tag an application manually, both before or after you run an application analysis.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Click the **Analysis** tab.
3. In the row of the required application, click the pen icon. The **Update application** window opens.
4. Select the desired tags from the **Select a tag(s)** drop-down list.
5. Click **Save**.

6.5.2. Automatic tagging

MTA can automatically add tags to the application based on the application analysis. Automatic tagging is especially useful when dealing with large portfolios of applications.

Automatic tagging of applications is enabled by default. You can disable automatic tagging by deselecting the **Enable automated tagging** checkbox in the **Advanced** section of the **Analysis configuration** wizard.

NOTE

To tag an application automatically, make sure that the **Enable automated tagging** checkbox is selected *before* you run an application analysis.

6.5.3. Displaying application tags

You can display the tags attached to a particular application.

NOTE

You can display the tags that were attached automatically only *after* you have run an application analysis.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Click the **Analysis** tab.
3. Click the name of the required application. A side pane opens.
4. Click the **Tags** tab. The tags attached to the application are displayed.

NOTE

You can filter the tags by source and tag category:

- The sources are **Analysis** and **Manual**.
- You can display the categories in a drop-down list. The categories are, for example, **HTTP**, **MVC**, **Web**, **Observability**, **Persistence**, **Application Type**, or **Data Center**.

6.5.4. Creating application tags

You can create custom tags for applications that MTA assesses or analyzes.

Procedure

1. In the **Migration** view, click **Controls**.
2. Click the **Tags** tab.
3. Click **Create tag**.
4. In the **Name** field in the opened dialogue, enter a unique name for the tag.
5. Click the **Tag category** field and select the category tag to associate with the tag.
6. Click **Create**.

6.5.5. Editing application tags

You can edit the defined application tags.

Procedure

1. In the **Migration** view, click **Controls**.
2. Click the **Tags** tab. A list of tag categories appears in the main pane.
3. Open a list of tags in the category by clicking the arrow to the left of the category name.

4. Select **Edit** from the drop-down menu.
5. Edit the tag's name in the **Name** field of the opened dialogue.
6. Click the **Tag category** field and select the category tag to associate with the tag.
7. Click **Save**.

6.5.6. Editing tag categories

You can edit the defined tag categories.

Procedure

1. In the **Migration** view, click **Controls**.
2. Click the **Tags** tab.
3. Select a defined tag category and click **Edit** to the right of the tag category.
4. Edit the tag category's name in the **Name** field.
5. Edit the category's **Rank** value.
6. Click the **Color** field and select a color for the tag category.
7. Click **Save**.

6.6. WORKING WITH ARCHETYPES

An **archetype** is a group of applications with common characteristics. You can use archetypes to assess multiple applications at once. By using archetypes, the Migration Toolkit for Applications (MTA) can apply questionnaires populated with questions that apply to common application characteristics.

Application archetypes are defined by criteria tags and the application taxonomy. Each archetype defines how the *assessment module* assesses the application according to the characteristics defined in that archetype. If the tags of an application match the criteria tags of an archetype, the application is associated with the archetype.

Creation of an archetype is defined by a series of **tags**, **stakeholders**, and **stakeholder groups**. The tags include the following types:

- **Criteria tags** are tags that the archetype requires to include an application as a member.

  NOTE

  If the archetype criteria tags match an application only partially, this application cannot be a member of the archetype. For example, if the application *a* only has tag *a*, but the archetype *a* criteria tags include tags *a* AND *b*, the application *a* will not be a member of the archetype *a*.

- **Archetype tags** are tags that are applied to the archetype entity.

6.6.1. Creating an archetype

When you create an archetype, an application in the inventory is automatically associated to that archetype if this application has the tags that match the tags of the archetype.

Procedure

1. Open the MTA web console.
2. In the left menu, click **Archetypes**.
3. Click **Create new archetype**.
4. In the form that opens, enter the following information for the new archetype:
   a. **Name**: A name of the new archetype (mandatory).
   b. **Description**: A description of the new archetype (optional).
   c. **Criteria Tags**: Tags that associate the assessed applications with the archetype (mandatory). If criteria tags are updated, the process to calculate the applications, which the archetype is associated with, is triggered again.
   d. **Archetype Tags**: Tags that the archetype assesses in the application (mandatory).
   e. **Stakeholder(s)**: Specific stakeholders involved in the application development and migration (optional).
   f. **Stakeholders Group(s)**: Groups of stakeholders involved in the application development and migration (optional).
5. Click **Create**. The new archetype appears in the list.

6.6.2. Archetype inheritance

Inheritance, which is the default setting, means that all applications associated with the archetype inherit the *assessment* and *review* from the archetype groups to which these applications belong. However, you can override inheritance for the application by completing an individual assessment and review.

6.6.3. Archetype assessment

An archetype is considered assessed when all required questionnaires have been answered.

An archetype is considered reviewed when it has been reviewed once even if multiple questionnaires have been marked as required.

If an application is associated with archetypes, this application is considered assessed when all associated archetypes have been assessed.

6.6.4. Deleting an archetype

Deleting an archetype deletes any associated assessment. All associated applications move to the **Unassessed** state.

6.7. ANALYZING AN APPLICATION

You can use the Migration Toolkit for Applications (MTA) user interface to configure and run an application analysis.

6.7.1. Configuring and running an application analysis

You can analyze more than one application at a time against more than one transformation target in the same analysis.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Click the **Analysis** tab.
3. Select an application that you want to analyze.
4. Review the credentials assigned to the application.
5. Click **Analyze**.
6. Select the **Analysis mode** from the list:
   - Binary.
   - Source code.
   - Source code and dependencies.
   - Upload a local binary. This option only appears if you are analyzing a single application.
7. If you choose **Upload a local binary**, you are prompted to **Upload a local binary**. Either drag and drop a file into the area provided or click **Upload** and select the file to upload.
8. Click **Next**.
9. Select one or more target options for the analysis:
   - Application server migration to either the following platforms:
     - JBoss EAP 7
     - JBoss EAP 8
   - Containerization
   - Quarkus
   - OracleJDK to OpenJDK
   - OpenJDK. Use this option to upgrade to either of the following JDK versions:
     - OpenJDK 11
     - OpenJDK 17
     - OpenJDK 21

- Linux. Use this option to ensure that there are no Microsoft Windows paths hard-coded into your applications.
- Jakarta EE 9. Use this option to migrate from Java EE 8.
- Spring Boot on Red Hat Runtimes
- Open Liberty
- Camel. Use this option to migrate from Apache Camel 2 to Apache Camel 3 or from Apache Camel 3 to Apache Camel 4.
- Azure
    - Azure App Service
    - Azure Kubernetes Service

10. Click **Next**.
11. Select one of the following **Scope** options to better focus the analysis:
    - Application and internal dependencies only.
    - Application and all dependencies, including known Open Source libraries.
    - Select the list of packages to be analyzed manually. If you choose this option, type the file name and click **Add**.
    - Exclude packages. If you choose this option, type the name of the package and click **Add**.
12. Click **Next**.
13. In **Advanced**, you can attach additional custom rules to the analysis by selecting the **Manual** or **Repository** mode:
    - In the **Manual** mode, click **Add Rules**. Drag and drop the relevant files or select the files from their directory and click **Add**.
    - In **Repository** mode, you can add rule files from a Git or Subversion repository.

    IMPORTANT

    Attaching custom rules is optional if you have already attached a migration target to the analysis. If you have not attached any migration target, you must attach rules.
14. Optional: Set any of the following options:
    - Target
    - Source(s)
    - Excluded rules tags. Rules with these tags are not processed. Add or delete as needed.
    - Enable automated tagging. Select the checkbox to automatically attach tags to the application. This checkbox is selected by default.

> NOTE
>
> Automatically attached tags are displayed only *after* you run the analysis.
>
> You can attach tags to the application manually instead of enabling automated tagging or in addition to it.

> NOTE
>
> Analysis engines use standard rules for a comprehensive set of migration targets, but if the target is not included or is a customized framework, custom rules can be added. Only manually uploaded custom rule files are validated.

15. Click **Next**.

16. In **Review**, verify the analysis parameters.

17. Click **Run**.
    The analysis status is **Scheduled** as MTA downloads the image for the container to execute. When the image is downloaded, the status changes to **In-progress.**

    > NOTE
    >
    > Analysis takes minutes to hours to run depending on the size of the application and the capacity and resources of the cluster.

    TIP

    MTA relies on Kubernetes scheduling capabilities to determine how many analyzer instances are created based on cluster capacity. If several applications are selected for analysis, by default, only one analyzer can be provisioned at a time. With more cluster capacity, more analysis processes can be executed in parallel.

18. When analysis is complete, click the **Report** link to see the results of the analysis.

6.7.2. Displaying the analysis details

You can view the details of an analysis only after you start running the analysis. If the status of an analysis is **Not started**, the **Analysis details** option is disabled.

Procedure

1. Click the Options menu (⋮).

2. Select **Analysis details**. The details are displayed in the **Analysis details for customers**window. You can choose either the YAML or JSON format.

6.7.3. Reviewing an analysis report

An MTA analysis report contains a number of sections, including a listing of the technologies used by the application, the dependencies of the application, and the lines of code that must be changed to successfully migrate or modernize the application.

For more information about the contents of an MTA analysis report, see Reviewing the reports.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Expand the application with a completed analysis.
3. Click **Reports**.
4. Click the dependencies or source links.
5. Click the tabs to review the report.

6.7.4. Downloading an analysis report

For your convenience, you can download analysis reports. By default, this option is disabled, but you can it enable by using the MTA user interface.

Procedure

1. In **Administration** view, click **General**.
2. Toggle the **Allow reports to be downloaded after running an analysis**switch.
3. Go to the **Migration** view and click **Application inventory**.
4. Open the page of the application for which you ran an analysis.
5. Click **Reports**.
6. Click either the **HTML** or **YAML** link:
    - By clicking the **HTML** link, you download the compressed **analysis-report-app-<application_name>.tar** file. Extracting this file creates a folder with the same name as the application.
    - By clicking the **YAML** link, you download the uncompressed **analysis-report-app-<application_name>.yaml** file.

6.8. CREATING CUSTOM MIGRATION TARGETS

Architects or users with **admin** permissions can create and maintain custom rulesets associated with custom migration targets. Architects can upload custom rule files and assign them to various custom migration targets. The custom migration targets can then be selected in the analysis configuration wizard.

By using ready-made custom migration targets, you can avoid configuring custom rules for each analysis run. This simplifies analysis configuration and execution for non-admin users or third-party developers.

Prerequisites

- You are logged in as a user with **admin** permissions.

Procedure

1. In the **Administration** view, click **Custom migration targets**.
2. Click **Create new**.
3. Enter the name and description of the target.
4. In the **Image** section, upload an image file for the target’s icon. The file can be in either PNG or JPEG format, up to 1 MB. If you do not upload any file, a default icon is used.
5. In the **Custom rules** section, select either **Upload manually** or **Retrieve from a repository**.
    - If you selected **Upload manually**, upload or drag and drop the required rule files from your local drive.
    - If you selected **Retrieve from a repository**, complete the following steps:
        i. Choose **Git** or **Subversion**.
        ii. Enter the **Source repository**, **Branch**, and **Root path** fields.
        iii. If the repository requires credentials, enter these credentials in the **Associated credentials** field.
6. Click **Create**.
   The new migration target appears on the **Custom migration targets** page. It can now be used by non-admin users in the **Migration** view.

CHAPTER 7. MANAGING APPLICATIONS WITH MTA

You can use the Migration Toolkit for Applications (MTA) user interface to perform the following tasks:

- Add applications.
- Assign application credentials.
- Import a list of applications.
- Download a CSV template for importing application lists.

7.1. APPLICATION ATTRIBUTES

You can add applications to the Migration Toolkit for Applications (MTA) user interface manually or by importing a list of applications.

MTA user interface applications have the following attributes:

- Name (free text)
- Description (optional, free text)
- Business service (optional, chosen from a list)
- Tags (optional, chosen from a list)
- Owner (optional, chosen from a list)
- Contributors (optional, chosen from a list)
- Source code (a path entered by the user)
- Binary (a path entered by the user)

7.2. ADDING AN APPLICATION

You can add an application to the **Application Inventory** for assessment and analysis.

TIP

Before creating an application, set up business services, check tags and tag categories, and create additions as needed.

Procedure

1. In the **Migration** view, click **Application Inventory**.
2. Click **Create new**.
3. Under **Basic information**, enter the following information or choose from a list:
    - Name
    - Description (optional)

   - Business service (optional)
   - Manual tags (optional, one or more)
   - Owner (optional)
   - Contributors (optional, one or more)
   - Comments (optional)

4. Click the arrow to the left of **Source Code**.
5. Enter the following information:
   - Repository type (**Git** or **Subversion**)
   - Source repository
   - Branch
   - Root path
6. Click the arrow to the left of **Binary**.
7. Enter the following information:
   - Group
   - Artifact
   - Version
   - Packaging
8. Click **Create**.

7.3. EDITING AN APPLICATION

You can edit an existing application in the **Application Inventory** and re-run an assessment or analysis for this application.

Prerequisites

- You are logged in to an MTA server.

Procedure

1. In the **Migration** view, click **Application Inventory**.
2. In the MTA user interface, select the **Migration** working mode.
3. Click **Application Inventory** in the left menu bar. A list of available applications appears in the main pane.
4. Click **Edit** (✎) to open the application settings.

5. Review the application settings. For a list of application settings, see Adding an application.
6. If you changed any application settings, click **Save**.

7.4. ASSIGNING CREDENTIALS TO AN APPLICATION

You can assign credentials to one or more applications.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Click the **Analysis** tab.
3. Click the Options menu (⋮) to the right of **Analyze** and select **Manage credentials**.
4. Select one credential from the **Source credentials** list and from the **Maven settings** list.
5. Click **Save**.

7.5. IMPORTING A LIST OF APPLICATIONS

You can import a **.csv** file that contains a list of applications and their attributes to the Migration Toolkit for Applications (MTA) user interface.

NOTE

Importing a list of applications does not overwrite any of the existing applications.

Procedure

1. Review the import file to ensure it contains all the required information in the required format.
2. In the **Migration** view, click **Application Inventory**.
3. Click the Options menu (⋮).
4. Click **Import**.
5. Select the desired file and click **Open**.
6. Optional: Select **Enable automatic creation of missing entities** This option is selected by default.
7. Verify that the import has completed and check the number of accepted or rejected rows.
8. Review the imported applications by clicking the arrow to the left of the checkbox.

   IMPORTANT

   Accepted rows might not match the number of applications in the Application inventory list because some rows are dependencies. To verify, check the **Record Type** column of the CSV file for applications defined as **1** and dependencies defined as **2**.

7.6. DOWNLOADING A CSV TEMPLATE

You can download a CSV template for importing application lists by using the Migration Toolkit for Applications (MTA) user interface.

Procedure

1. In the **Migration** view, click **Application inventory**.
2. Click the Options menu (⋮) to the right of **Review**.
3. Click **Manage imports** to open the **Application imports** page.
4. Click the Options menu (⋮) to the right of **Import**.
5. Click **Download CSV template**.

7.7. CREATING A MIGRATION WAVE

By creating a migration wave, you can group applications that you want to migrate on a given schedule. You can also track each migration by exporting a list of the wave’s applications to the Jira issue management system. This automatically creates a separate Jira issue for each application of the migration wave.

Procedure

1. In the **Migration** view, click **Migration waves**.
2. Click **Create new**. The **New migration wave** window opens.
3. Enter the following information or select from drop-down lists:
   - Name (optional). If the name is not given, you can use the start and end dates to identify migration waves.
   - Potential start date. This date must be later than the current date.
   - Potential end date. This date must be later than the start date.
   - Stakeholders (optional)
   - Stakeholder groups (optional)
4. Click **Create**. The new migration wave appears in the list of existing migration waves.
5. To assign applications to the migration wave, click the Options menu (⋮) to the right of the migration wave and select **Manage applications**.
   The **Manage applications** window opens that displays the list of applications that are not assigned to any other migration wave.
6. Select the checkboxes of the applications that you want to assign to the migration wave.
7. Click **Save**.

NOTE

The owner and the contributors of each application associated with the migration wave are automatically added to the migration wave's list of stakeholders.

8. Optional: To update a migration wave, select **Update** from the migration wave's **Options** menu (three dots). The **Update migration wave** window opens.

7.8. CREATING JIRA ISSUES FOR A MIGRATION WAVE

You can use a migration wave to create Jira issues automatically for each application assigned to the migration wave. A separate Jira issue is created for each application associated with the migration wave. The following fields of each issue are filled in automatically:

- Title: **Migrate <application name>**
- Reporter: Username of the token owner.
- Description: **Created by Konveyor**

NOTE

You cannot delete an application if it is linked to a Jira ticket or is associated with a migration wave. To unlink the application from the Jira ticket, click the **Unlink from Jira** icon in the details view of the application or in the details view of a migration wave.

Procedure

1. In the **Migration** view, click **Migration waves**.
2. Click the Options menu (⋮) to the right of the migration wave for which you want to create Jira issues and select **Export to Issue Manager**. The **Export to Issue Manager** window opens.
3. Select the Jira Cloud or Jira Server/Datacenter instance type.
4. Select the instance, project, and issue type from the lists.
5. Click **Export**. The status of the migration wave on the **Migration waves** page changes to **Issues Created**.
6. Optional: To see the status of each individual application of a migration wave, click the **Status** column.
7. Optional: To see if any particular application is associated with a migration wave, open the application's **Details** tab on the **Application inventory** page.

Revised on 2024-05-03 10:40:41 UTC