Red Hat Decision Manager 7.5

Deploying a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform

Last Updated: 2021-03-30

Red Hat Decision Manager 7.5 Deploying a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform

Red Hat Customer Content Services
brms-docs@redhat.com

Legal Notice

Abstract

This document describes how to deploy a Red Hat Decision Manager 7.5 immutable server environment on Red Hat OpenShift Container Platform.

Table of Contents

PREFACE

As a system engineer, you can deploy a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform to provide an infrastructure to execute services and other business assets. You can use standard integration tools to manage the immutable Decision Server image. You can create new server images to add and update the business assets.

Prerequisites

- Red Hat OpenShift Container Platform version 3.11 is deployed.
- At least two gigabytes of memory are available in the OpenShift cluster/namespace.
- The OpenShift project for the deployment has been created.
- You are logged in to the project using the **oc** command. For more information about the **oc** command-line tool, see the OpenShift CLI Reference. If you want to use the OpenShift Web console to deploy templates, you must also be logged on using the Web console.

NOTE

Since Red Hat Decision Manager version 7.5, support for Red Hat OpenShift Container Platform 3.x is deprecated, including installation using all templates and using the Automation Broker (Ansible Playbook). New features might not be added, and this functionality will be removed in a future release.

CHAPTER 1. OVERVIEW OF RED HAT DECISION MANAGER ON RED HAT OPENSHIFT CONTAINER PLATFORM

You can deploy Red Hat Decision Manager into a Red Hat OpenShift Container Platform environment.

In this solution, components of Red Hat Decision Manager are deployed as separate OpenShift pods. You can scale each of the pods up and down individually to provide as few or as many containers as required for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.

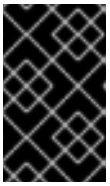
The following key components of Red Hat Decision Manager are available on OpenShift:

- Decision Server, also known as *Execution Server* or *KIE Server*, is the infrastructure element that runs decision services and other deployable assets (collectively referred to as *services*) . All logic of the services runs on execution servers.
  You can scale up a Decision Server pod to provide as many copies as required, running on the same host or different hosts. As you scale a pod up or down, all of its copies run the same services. OpenShift provides load balancing and a request can be handled by any of the pods.

  You can deploy a separate Decision Server pod to run a different group of services. That pod can also be scaled up or down. You can have as many separate replicated Decision Server pods as required.

- Business Central is a web-based interactive environment used for authoring services. It also provides a management console. You can use Business Central to develop services and deploy them to Decision Servers.
  Business Central is a centralized application. However, you can configure it for high availability, where multiple pods run and share the same data.

  Business Central includes a Git repository that holds the source for the services that you develop on it. It also includes a built-in Maven repository. Depending on configuration, Business Central can place the compiled services (KJAR files) into the built-in Maven repository or (if configured) into an external Maven repository.



IMPORTANT

In the current version, high-availability Business Central functionality is for Technology Preview only. For more information on Red Hat Technology Preview features, see Technology Preview Features Scope .

You can arrange these and other components into various environment configurations within OpenShift.

The following environment types are typical:

- *Authoring or managed environment* : An environment architecture that can be used for creating and modifying services using Business Central and also for running services on Decision Servers. It consists of pods that provide Business Central for the authoring work and one or more Decision Servers for execution of the services. Each Decision Server is a pod that you can replicate by scaling it up or down as necessary. You can deploy and undeploy services on each Decision Server using Business Central. For instructions about deploying this environment, see *Deploying a Red Hat Decision Manager authoring or managed server environment on Red Hat OpenShift Container Platform*.

- *Deployment with immutable servers* : An alternate environment for running existing services for staging and production purposes. In this environment, when you deploy a Decision Server pod, it

builds an image that loads and starts a service or group of services. You cannot stop any service on the pod or add any new service to the pod. If you want to use another version of a service or modify the configuration in any other way, you deploy a new server image and displace the old one. In this system, the Decision Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows and do not need to use any other tools to manage the pods. For instructions about deploying this environment, see *Deploying a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform*.

You can also deploy a *trial* or evaluation environment. This environment includes Business Central and a Decision Server. You can set it up quickly and use it to evaluate or demonstrate developing and running assets. However, the environment does not use any persistent storage, and any work you do in the environment is not saved. For instructions about deploying this environment, see *Deploying a Red Hat Decision Manager trial environment on Red Hat OpenShift Container Platform*.

To deploy a Red Hat Decision Manager environment on OpenShift, you can use the templates that are provided with Red Hat Decision Manager.

CHAPTER 2. PREPARING TO DEPLOY RED HAT DECISION MANAGER IN YOUR OPENSHIFT ENVIRONMENT

Before deploying Red Hat Decision Manager in your OpenShift environment, you must complete several tasks. You do not need to repeat these tasks if you want to deploy additional images, for example, for new versions of decision services or for other decision services

2.1. ENSURING THE AVAILABILITY OF IMAGE STREAMS AND THE IMAGE REGISTRY

To deploy Red Hat Decision Manager components on Red Hat OpenShift Container Platform, you must ensure that OpenShift can download the correct images from the Red Hat registry. To download the images, OpenShift requires *image streams*, which contain the information about the location of images. OpenShift also must be configured to authenticate with the Red Hat registry using your service account user name and password.

Some versions of the OpenShift environment include the required image streams. You must check if they are available. If image streams are available in OpenShift by default, you can use them if the OpenShift infrastructure is configured for registry authentication server. The administrator must complete the registry authentication configuration when installing the OpenShift environment.

Otherwise, you can configure registry authentication in your own project and install the image streams in that project.

Procedure

1. Determine whether Red Hat OpenShift Container Platform is configured with the user name and password for Red Hat registry access. For details about the required configuration, see Configuring a Registry Location. If you are using an OpenShift Online subscription, it is configured for Red Hat registry access.

2. If Red Hat OpenShift Container Platform is configured with the user name and password for Red Hat registry access, enter the following commands:

```
$ oc get imagestreamtag -n openshift | grep rhdm75-decisioncentral-openshift
$ oc get imagestreamtag -n openshift | grep rhdm75-kieserver-openshift
```

   If the outputs of both commands are not empty, the required image streams are available in the **openshift** namespace and no further action is required.

3. If the output of one or both of the commands is empty or if OpenShift is not configured with the user name and password for Red Hat registry access, complete the following steps:

   a. Ensure you are logged in to OpenShift with the **oc** command and that your project is active.

   b. Complete the steps documented in Registry Service Accounts for Shared Environments. You must log in to the Red Hat Customer Portal to access the document and to complete the steps to create a registry service account.

   c. Select the **OpenShift Secret** tab and click the link under **Download secret** to download the YAML secret file.

   d. View the downloaded file and note the name that is listed in the **name:** entry.

   e. Enter the following commands:

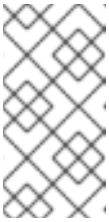
```
oc create -f <file_name>.yaml
oc secrets link default <secret_name> --for=pull
oc secrets link builder <secret_name> --for=pull
```

Replace **<file_name>** with the name of the downloaded file and **<secret_name>** with the name that is listed in the **name:** entry of the file.

f. Download the **rhdm-7.5.1-openshift-templates.zip** product deliverable file from the Software Downloads page and extract the **rhdm75-image-streams.yaml** file.

g. Enter the following command:

```
$ oc apply -f rhdm75-image-streams.yaml
```



NOTE

If you complete these steps, you install the image streams into the namespace of your project. In this case, when you deploy the templates, you must set the **IMAGE_STREAM_NAMESPACE** parameter to the name of this project.

2.2. CREATING THE SECRETS FOR DECISION SERVER

OpenShift uses objects called *secrets* to hold sensitive information such as passwords or keystores. For more information about OpenShift secrets, see the Secrets chapter in the OpenShift documentation.

You must create an SSL certificate for HTTP access to Decision Server and provide it to your OpenShift environment as a secret.

Procedure

1. Generate an SSL keystore with a private and public key for SSL encryption for Decision Server. For more information on how to create a keystore with self-signed or purchased SSL certificates, see Generate a SSL Encryption Key and Certificate .

NOTE

In a production environment, generate a valid signed certificate that matches the expected URL for Decision Server.

2. Save the keystore in a file named **keystore.jks**.
3. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.
4. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.
5. Use the **oc** command to generate a secret named **kieserver-app-secret** from the new keystore file:

```
$ oc create secret generic kieserver-app-secret --from-file=keystore.jks
```

2.3. EXTRACTING THE SOURCE CODE FROM BUSINESS CENTRAL FOR USE IN AN S2I BUILD

If you are using Business Central for authoring services, you can extract the source code for your service and place it into a separate Git repository, such as GitHub or an on-premise installation of GitLab, for use in the S2I build.

Procedure

1. Use the following command to extract the source code:

```
git clone https://<decision-central-host>:443/git/<MySpace>/<MyProject>
```

In this command, replace the following variables:

- **<decision-central-host>** with the host on which Business Central is running
- **<MySpace>** with the name of the Business Central space in which the project is located
- **<MyProject>** with the name of the project

NOTE

To view the full Git URL for a project in Business Central, click **Menu → Design → *<MyProject>* → Settings**.

NOTE

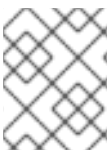
If you are using self-signed certificates for HTTPS communication, the command might fail with an **SSL certificate problem** error message. In this case, disable SSL certificate verification in **git**, for example, using the **GIT_SSL_NO_VERIFY** environment variable:

```
env GIT_SSL_NO_VERIFY=true git clone https://<decision-central-host>:443/git/<MySpace>/<MyProject>
```

2. Upload the source code to another Git repository, such as GitHub or GitLab, for the S2I build.

2.4. PREPARING A MAVEN MIRROR REPOSITORY FOR OFFLINE USE

If your Red Hat OpenShift Container Platform environment does not have outgoing access to the public Internet, you must prepare a Maven repository with a mirror of all the necessary artifacts and make this repository available to your environment.



NOTE

You do not need to complete this procedure if your Red Hat OpenShift Container Platform environment is connected to the Internet.

Prerequisites

- A computer that has outgoing access to the public Internet is available.

Procedure

1. Prepare a Maven release repository to which you can write. The repository must allow read access without authentication. Your OpenShift environment must have access to this repository. You can deploy a Nexus repository manager in the OpenShift environment. For instructions about setting up Nexus on OpenShift, see Setting up Nexus. Use this repository as a mirror repository. If you are planning to create immutable servers from KJAR services, place your services in this repository as well. You must configure this repository as the external Maven repository. You cannot configure a separate mirror repository in an immutable environment.

2. On the computer that has an outgoing connection to the public Internet, complete the following steps:

   a. Download the latest version of the Offliner tool.

   b. Download the **rhdm-7.5.1-offliner.txt** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

   c. Enter the following command to use the Offliner tool to download the required artifacts:

   ```
   java -jar offliner-<version>.jar -r https://maven.repository.redhat.com/ga/ -r
   https://repo1.maven.org/maven2/ -d /home/user/temp rhdm-7.5.1-offliner.txt
   ```

   Replace **/home/user/temp** with an empty temporary directory and **<version>** with the version of the Offliner tool that you downloaded. The download can take a significant amount of time.

   d. Upload all artifacts from the temporary directory to the Maven mirror repository that you prepared. You can use the Maven Repository Provisioner utility to upload the artifacts.

3. If you developed services outside Business Central and they have additional dependencies, add the dependencies to the mirror repository. If you developed the services as Maven projects, you can use the following steps to prepare these dependencies automatically. Complete the steps on the computer that has an outgoing connection to the public Internet.

   a. Create a backup of the local Maven cache directory (**~/.m2/repository**) and then clear the directory.

   b. Build the source of your projects using the **mvn clean install** command.

   c. For every project, enter the following command to ensure that Maven downloads all runtime dependencies for all the artifacts generated by the project:

   ```
   mvn -e -DskipTests dependency:go-offline -f /path/to/project/pom.xml --batch-mode -
   Djava.net.preferIPv4Stack=true
   ```

   Replace **/path/to/project/pom.xml** with the correct path to the **pom.xml** file of the project.

   d. Upload all artifacts from the local Maven cache directory (**~/.m2/repository**) to the Maven mirror repository that you prepared. You can use the Maven Repository Provisioner utility to upload the artifacts.

CHAPTER 3. ENVIRONMENT WITH IMMUTABLE SERVERS

You can deploy an environment that includes one or more pods running *immutable* Decision Server with preloaded services. Each Decision Server pod can be separately scaled as necessary.

On an immutable Decision Server, any services must be loaded onto the server at the time the image is created. You cannot deploy or undeploy services on a running immutable Decision Server. The advantage of this approach is that the Decision Server with the services in it runs like any other containerized service and does not require specialized management. The Decision Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows as necessary.

When you create a Decision Server image, you can build your services using S2I (Source to Image). Provide a Git repository with the source of your services and other business assets; if you develop the services or assets in Business Central, copy the source into a separate repository for the S2I build. OpenShift automatically builds the source, installs the services into the Decision Server image, and starts the containers with the services.

If you are using Business Central for authoring services, you can extract the source for your process and place it into a separate Git repository (such as GitHub or an on-premise installation of GitLab) for use in the S2I build.

Alternatively, you can create a similar Decision Server deployment using services that are already built as KJAR files. In this case, you must provide the services in a Maven repository. You can use the built-in repository of the Business Central or your own repository (for example, a Nexus deployment). When the server pod starts, it retrieves the KJAR services from the Maven repository. Services on the pod are never updated or changed. At every restart or scaling of the pod, the server retrieves the files from the repository, so you must ensure they do not change on the Maven repository to keep the deployment immutable.

With both methods of creating immutable images, no further management of the image is required. If you want to use a new version of a service, you can build a new image.

3.1. DEPLOYING AN IMMUTABLE DECISION SERVER USING AN S2I BUILD

You can deploy an immutable Decision Server using an S2I build. When you deploy the server, the deployment procedure retrieves the source code for any services that must run on this server, builds the services, and includes them in the server image.

You cannot deploy or undeploy services on a running immutable Decision Server. You can use Business Central to view monitoring information. The Decision Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows as necessary.

You can enable JMS capabilities of the immutable Decision Server. With JMS capabilities you can interact with the server through JMS API using an external AMQ message broker.

If a Business Central is deployed in the same namespace, it discovers the immutable Decision Server automatically. You can use Business Central to start and stop (but not deploy) services on the immutable Decision Server.

3.1.1. Starting configuration of the template for an immutable Decision Server using S2I

To deploy an immutable Decision Server using an S2I build, use the **rhdm75-prod-immutable-kieserver-amq.yaml** template file if you want to enable JMS capabilities. Otherwise, use the **rhdm75-prod-immutable-kieserver.yaml** template file.

Procedure

1. Download the **rhdm-7.5.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
2. Extract the required template file.
3. Use one of the following methods to start deploying the template:
   - To use the OpenShift Web UI, in the OpenShift application console select **Add to Project → Import YAML / JSON** and then select or paste the **<template-file-name>.yaml** file. In the **Add Template** window, ensure **Process the template** is selected and click **Continue**.
   - To use the OpenShift command line console, prepare the following command line:

     ```
     oc new-app -f <template-path>/<template-file-name>.yaml -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
     ```

     In this command line, make the following changes:
     - Replace **<template-path>** with the path to the downloaded template file.
     - Replace **<template-file-name>** with the name of the template file.
     - Use as many **-p PARAMETER=value** pairs as needed to set the required parameters.

Next steps

Set the parameters for the template. Follow the steps in Section 3.1.2, "Setting required parameters for an immutable Decision Server using S2I" to set common parameters. You can view the template file to see descriptions for all parameters.

3.1.2. Setting required parameters for an immutable Decision Server using S2I

When configuring the template to deploy an immutable Decision Server using an S2I build, you must set the following parameters in all cases.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

1. Set the following parameters:
   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for Decision Server, as created in Section 2.2, "Creating the secrets for Decision Server".
   - **KIE Server Certificate Name**(**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 2.2, "Creating the secrets for Decision Server".

- **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 2.2, "Creating the secrets for Decision Server" .
- **Application Name** (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and Decision Server. OpenShift uses the application name to create a separate set of deployment configurations, services, routes, labels, and artifacts. You can deploy several applications using the same template into the same project, as long as you use different application names. Also, the application name determines the name of the server configuration (server template) that the Decision Server joins on Business Central. If you are deploying several Decision Servers, you must ensure each of the servers has a different application name.
- **KIE Server Container Deployment**(**KIE_SERVER_CONTAINER_DEPLOYMENT**): The identifying information of the decision service (KJAR file) that the deployment must pull from the local or external repository after building your source. The format is **<containerId>=<groupId>:<artifactId>:<version>** or, if you want to specify an alias name for the container, **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>**. You can provide two or more KJAR files using the | separator, as illustrated in the following example:

  ```
  containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2
  ```

  To avoid duplicate container IDs, the artifact ID must be unique for each artifact built or used in your project.
- **Git Repository URL** (**SOURCE_REPOSITORY_URL**): The URL for the Git repository that contains the source for your services.
- **Git Reference** (**SOURCE_REPOSITORY_REF**): The branch in the Git repository.
- **Context Directory** (**CONTEXT_DIR**): The path to the source within the project downloaded from the Git repository.
- **Artifact Directory** (**ARTIFACT_DIR**): The path within the project that contains the required binary files (KJAR files and any other necessary files) after a successful Maven build. Normally this directory is the target directory of the build. However, you can provide prebuilt binaries in this directory in the Git repository.
- **ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**): The namespace where the image streams are available. If the image streams were already available in your OpenShift environment (see Section 2.1, "Ensuring the availability of image streams and the image registry"), the namespace is **openshift**. If you have installed the image streams file, the namespace is the name of the OpenShift project.

2. You can set the following user name and password. By default, the deployment automatically generates the password.
   - **KIE Server User** (**KIE_SERVER_USER**) and **KIE Server Password** (**KIE_SERVER_PWD**): The user name and password that a client application can use to connect to any of the Decision Servers.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.3. Configuring the image stream namespace for an immutable Decision Server using S2I

If you created image streams in a namespace that is not **openshift**, you must configure the namespace in the template.

If all image streams were already available in your Red Hat OpenShift Container Platform environment, you can skip this procedure.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

If you installed an image streams file according to instructions in Section 2.1, "Ensuring the availability of image streams and the image registry", set the **ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) parameter to the name of your OpenShift project.

3.1.4. Configuring information about a Business Central instance for an immutable Decision Server using S2I

If you want to enable a connection from a Business Central instance in the same namespace to the Decision Server, you must configure information about the Business Central instance.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

1. Set the following parameters:
   - **KIE Admin User** (**KIE_ADMIN_USER**) and **KIE Admin Password** (**KIE_ADMIN_PWD**): The user name and password for the administrative user. These values must be the same as the **KIE_ADMIN_USER** and **KIE_ADMIN_PWD** settings for the Business Central. If the Business Central uses RH-SSO or LDAP authentication, these values must be a user name and password configured in the authentication system with an administrator role for the Business Central.
   - **Name of the Business Central service**(**DECISION_CENTRAL_SERVICE**): The OpenShift service name for the Business Central.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.5. Setting an optional Maven repository for an immutable Decision Server using S2I

When configuring the template to deploy an immutable Decision Server using an S2I build, if your source build includes dependencies that are not available on the public Maven tree and require a separate custom Maven repository, you must set parameters to access the repository.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

To configure access to a custom Maven repository, set the following parameters:

- **Maven repository URL** (**MAVEN_REPO_URL**): The URL for the Maven repository.
- **Maven repository ID** (**MAVEN_REPO_ID**): An identifier for the Maven repository. The default value is **repo-custom**.
- **Maven repository username** (**MAVEN_REPO_USERNAME**): The username for the Maven repository.
- **Maven repository password** (**MAVEN_REPO_PASSWORD**): The password for the Maven repository.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.6. Configuring access to a Maven mirror in an environment without a connection to the public Internet for an immutable Decision Server using S2I

When configuring the template to deploy an immutable Decision Server using an S2I build, if your OpenShift environment does not have a connection to the public Internet, you must configure access to a Maven mirror that you set up according to Section 2.4, "Preparing a Maven mirror repository for offline use".

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

To configure access to the Maven mirror, set the following parameters:

- **Maven mirror URL** (**MAVEN_MIRROR_URL**): The URL for the Maven mirror repository that you set up in Section 2.4, "Preparing a Maven mirror repository for offline use" . This URL must be accessible from a pod in your OpenShift environment.
- **Maven mirror of** (**MAVEN_MIRROR_OF**): The value that determines which artifacts are to be retrieved from the mirror. For instructions about setting the **mirrorOf** value, see Mirror Settings in the Apache Maven documentation. The default value is **external:***. With this value, Maven retrieves every required artifact from the mirror and does not query any other repositories.

- If you configure an external Maven repository (**MAVEN_REPO_URL**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror, for example, **external:*,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.
- If you configure a built-in Business Central Maven repository (**BUSINESS_CENTRAL_MAVEN_SERVICE**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror: **external:*,!repo-rhdmcentr**.
- If you configure both repositories, change **MAVEN_MIRROR_OF** to exclude the artifacts in both repositories from the mirror: **external:*,!repo-rhdmcentr,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.7. Configuring communication with an AMQ server for an immutable Decision Server using S2I

If you use the **rhdm75-prod-immutable-kieserver-amq.yaml** template file, JMS capabilities of the Decision Server are enabled. You can interact with the server through JMS API, using an external AMQ message broker.

If necessary for your environment, you can modify the JMS configuration.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I", using the **rhdm75-prod-immutable-kieserver-amq.yaml** template file.

Procedure

Set any of the following parameters as required for your environment:

- **AMQ Username** (**AMQ_USERNAME**) and **AMQ Password** (**AMQ_PASSWORD**): The user name and password of a standard broker user, if user authentication in the broker is required in your environment.
- **AMQ Role** (**AMQ_ROLE**): The user role for the standard broker user. The default role is **admin**.
- **AMQ Queues** (**AMQ_QUEUES**): AMQ queue names, separated by commas. These queues are automatically created when the broker starts and are accessible as JNDI resources in the JBoss EAP server. If you use custom queue names, you must also set the same queue names in the **KIE_SERVER_JMS_QUEUE_RESPONSE**, **KIE_SERVER_JMS_QUEUE_REQUEST**, **KIE_SERVER_JMS_QUEUE_SIGNAL**, **KIE_SERVER_JMS_QUEUE_AUDIT**, and **KIE_SERVER_JMS_QUEUE_EXECUTOR** parameters.
- **AMQ Global Max Size** (**AMQ_GLOBAL_MAX_SIZE**): The maximum amount of memory that message data can consume. If no value is specified, half of the memory available in the pod is allocated.

- **AMQ Protocols** (**AMQ_PROTOCOL**): Broker protocols that the Decision Server can use to communicate with the AMQ server, separated by commas. Allowed values are **openwire**, **amqp**, **stomp**, and **mqtt**. Only **openwire** is supported by JBoss EAP. The default value is **openwire**.
- **AMQ Broker Image** (**AMQ_BROKER_IMAGESTREAM_NAME**): The image stream name for the AMQ broker image.

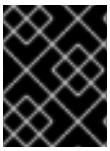
Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.8. Setting parameters for RH-SSO authentication for an immutable Decision Server using S2I

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy an immutable Decision Server using an S2I build.



IMPORTANT

Do not configure LDAP authentication and RH-SSO authentication in the same deployment.

Prerequisites

- A realm for Red Hat Decision Manager is created in the RH-SSO authentication system.
- User names and passwords for Red Hat Decision Manager are created in the RH-SSO authentication system. For a list of the available roles, see Chapter 4, *Red Hat Decision Manager roles and users*. In order to set the parameters for the environment, an administrative user with the **kie-server,rest-all,admin** roles is required. The default user name for this user is **adminUser**. This user can administer and use the environment.
- Clients are created in the RH-SSO authentication system for all components of the Red Hat Decision Manager environment that you are deploying. The client setup contains the URLs for the components. You can review and edit the URLs after deploying the environment. Alternatively, the Red Hat Decision Manager deployment can create the clients. However, this option provides less detailed control over the environment.
- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

1. Set the **KIE_ADMIN_USER** and **KIE_ADMIN_PASSWORD** parameters of the template to the user name and password of the administrative user that you created in the RH-SSO authentication system.
2. Set the following parameters:
   - **RH-SSO URL** (**SSO_URL**): The URL for RH-SSO.
   - **RH-SSO Realm name** (**SSO_REALM**): The RH-SSO realm for Red Hat Decision Manager.

- **RH-SSO Disable SSL Certificate Validation** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): Set to **true** if your RH-SSO installation does not use a valid HTTPS certificate.

3. Complete one of the following procedures:

   a. If you created the client for Red Hat Decision Manager within RH-SSO, set the following parameters in the template:

      - **Business Central RH-SSO Client name** (**DECISION_CENTRAL_SSO_CLIENT**): The RH-SSO client name for Business Central.
      - **KIE Server RH-SSO Client name** (**KIE_SERVER_SSO_CLIENT**): The RH-SSO client name for Decision Server.
      - **KIE Server RH-SSO Client Secret** (**KIE_SERVER_SSO_SECRET**): The secret string that is set in RH-SSO for the client for Decision Server.

   b. To create the clients for Red Hat Decision Manager within RH-SSO, set the following parameters in the template:

      - **KIE Server RH-SSO Client name** (**KIE_SERVER_SSO_CLIENT**): The name of the client to create in RH-SSO for Decision Server.
      - **KIE Server RH-SSO Client Secret** (**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for Decision Server.
      - **RH-SSO Realm Admin Username** (**SSO_USERNAME**) and **RH-SSO Realm Admin Password** (**SSO_PASSWORD**): The user name and password for the realm administrator user for the RH-SSO realm for Red Hat Decision Manager. You must provide this user name and password in order to create the required clients.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, “Completing deployment of the template for an immutable Decision Server using S2I”.

After completing the deployment, review the URLs for components of Red Hat Decision Manager in the RH-SSO authentication system to ensure they are correct.

3.1.9. Setting parameters for LDAP authentication for an immutable Decision Server using S2I

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an immutable Decision Server using an S2I build.

IMPORTANT

Do not configure LDAP authentication and RH-SSO authentication in the same deployment.

Prerequisites

- You created user names and passwords for Red Hat Decision Manager in the LDAP system. For a list of the available roles, see Chapter 4, *Red Hat Decision Manager roles and users* . As a minimum, in order to set the parameters for the environment, you created the following users:
  - An administrative user with the **kie-server,rest-all,admin** roles. This user can administer and use the environment.
  - A server user with the **kie-server,rest-all,user** roles. This user can make REST API calls to the Decision Server.
- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

1. In the LDAP service, create all user names in the deployment parameters. If you do not set any of the parameters, create users with the default user names. The created users must also be assigned to roles:
   - **KIE_ADMIN_USER**: default user name **adminUser**, roles: **kie-server,rest-all,admin**
   - **KIE_SERVER_USER**: default user name **executionUser**, roles **kie-server,rest-all,guest**
     For the user roles that you can configure in LDAP, see Roles and users.
2. Set the **AUTH_LDAP*** parameters of the template. These parameters correspond to the settings of the **LdapExtended** Login module of Red Hat JBoss EAP. For instructions about using these settings, see LdapExtended login module .
   If the LDAP server does not define all the roles required for your deployment, you can map LDAP groups to Red Hat Decision Manager roles. To enable LDAP role mapping, set the following parameters:
   - **RoleMapping rolesProperties file path** (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): The fully qualified path name of a file that defines role mapping, for example, **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**. You must provide this file and mount it at this path in all applicable deployment configurations; for instructions, see Section 3.3, "(Optional) Providing the LDAP role mapping file" .
   - **RoleMapping replaceRole property** (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): If set to **true**, mapped roles replace the roles defined on the LDAP server; if set to **false**, both mapped roles and roles defined on the LDAP server are set as user application roles. The default setting is **false**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

3.1.10. Enabling Prometheus metric collection for an immutable Decision Server using S2I

If you want to configure your Decision Server deployment to use Prometheus to collect and store metrics, enable support for this feature in Decision Server at deployment time.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for an immutable Decision Server using S2I".

Procedure

To enable support for Prometheus metric collection, set the **Prometheus Server Extension Disabled** (**PROMETHEUS_SERVER_EXT_DISABLED**) parameter to **false**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.11, "Completing deployment of the template for an immutable Decision Server using S2I".

For instructions about configuring Prometheus metrics collection, see *Managing and monitoring Decision Server*.

3.1.11. Completing deployment of the template for an immutable Decision Server using S2I

After setting all the required parameters in the OpenShift Web UI or in the command line, complete deployment of the template.

Procedure

Depending on the method that you are using, complete the following steps:

- In the OpenShift Web UI, click **Create**.
  - If the **This will create resources that may have security or project behavior implications** message appears, click **Create Anyway**.
- Complete the command line and press Enter.

3.2. DEPLOYING AN IMMUTABLE DECISION SERVER FROM KJAR SERVICES

You can deploy an immutable Decision Server using services that are already built as KJAR files.

You must provide the services in a Maven repository. You can use the built-in repository of the Business Central or your own repository (for example, a Nexus deployment). When the server pod starts, it retrieves the KJAR services from the Maven repository. Services on the pod are never updated or changed. At every restart or scaling of the pod, the server retrieves the files from the repository, so you must ensure they do not change on the Maven repository to keep the deployment immutable.

You cannot deploy or undeploy services on a running immutable Decision Server. You can use Business Central to view monitoring information. The Decision Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows as necessary.

If a Business Central is deployed in the same namespace, it discovers the immutable Decision Server automatically. You can use Business Central to start and stop (but not deploy) services on the immutable Decision Server and to view monitoring data.

3.2.1. Starting configuration of the template for an immutable Decision Server from KJAR services

To deploy an immutable Decision Server from KJAR services, use the **rhdm75-kieserver.yaml** template file.

Procedure

1. Download the **rhdm-7.5.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
2. Extract the **rhdm75-kieserver.yaml** template file.
3. Use one of the following methods to start deploying the template:
   - To use the OpenShift Web UI, in the OpenShift application console select **Add to Project → Import YAML / JSON** and then select or paste the **rhdm75-kieserver.yaml** file. In the **Add Template** window, ensure **Process the template** is selected and click **Continue**.
   - To use the OpenShift command line console, prepare the following command line:

     ```
     oc new-app -f <template-path>/rhdm75-kieserver.yaml -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p PARAMETER=value
     ```

     In this command line, make the following changes:
     - Replace **<template-path>** with the path to the downloaded template file.
     - Use as many **-p PARAMETER=value** pairs as needed to set the required parameters.

Next steps

Set the parameters for the template. Follow the steps in Section 3.2.2, "Setting required parameters for an immutable Decision Server from KJAR services" to set common parameters. You can view the template file to see descriptions for all parameters.

3.2.2. Setting required parameters for an immutable Decision Server from KJAR services

When configuring the template to deploy an immutable Decision Server from KJAR services, you must set the following parameters in all cases.

Prerequisites

- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

1. Set the following parameters:
   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for Decision Server, as created in Section 2.2, "Creating the secrets for Decision Server".

- **KIE Server Certificate Name** (**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 2.2, "Creating the secrets for Decision Server".
- **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 2.2, "Creating the secrets for Decision Server".
- **Application Name** (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and Decision Server. OpenShift uses the application name to create a separate set of deployment configurations, services, routes, labels, and artifacts. You can deploy several applications using the same template into the same project, as long as you use different application names. Also, the application name determines the name of the server configuration (server template) that the Decision Server joins on Business Central. If you are deploying several Decision Servers, you must ensure each of the servers has a different application name.
- **Maven repository URL** (**MAVEN_REPO_URL**): A URL for a Maven repository. You must upload all the processes (KJAR files) that are to be deployed on the Decision Server into this repository.
- **Maven repository ID** (**MAVEN_REPO_ID**): An identifier for the Maven repository. The default value is **repo-custom**.
- **Maven repository username** (**MAVEN_REPO_USERNAME**): The username for the Maven repository.
- **Maven repository password** (**MAVEN_REPO_PASSWORD**): The password for the Maven repository.
- **KIE Server Container Deployment** (**KIE_SERVER_CONTAINER_DEPLOYMENT**): The identifying information of the decision services (KJAR files) that the deployment must pull from the Maven repository. The format is **<containerId>=<groupId>:<artifactId>:<version>** or, if you want to specify an alias name for the container, **<containerId>(<aliasId>)=<groupId>:<artifactId>:<version>**. You can provide two or more KJAR files using the | separator, as illustrated in the following example:

  ```
  containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2
  ```

- **KIE Server Mode** (**KIE_SERVER_MODE**): In the **rhdm75-kieserver-*.yaml** templates the default value is **PRODUCTION**. In **PRODUCTION** mode, you cannot deploy **SNAPSHOT** versions of KJAR artifacts on the Decision Server and cannot change versions of an artifact in an existing container. To deploy a new version with **PRODUCTION** mode, create a new container on the same Decision Server. To deploy **SNAPSHOT** versions or to change versions of an artifact in an existing container, set this parameter to **DEVELOPMENT**.
- **ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**): The namespace where the image streams are available. If the image streams were already available in your OpenShift environment (see Section 2.1, "Ensuring the availability of image streams and the image registry"), the namespace is **openshift**. If you have installed the image streams file, the namespace is the name of the OpenShift project.

2. You can set the following user name and password. By default, the deployment automatically generates the password.
   - **KIE Server User** (**KIE_SERVER_USER**) and **KIE Server Password** (**KIE_SERVER_PWD**): The user name and password that a client application can use to connect to any of the Decision Servers.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

3.2.3. Configuring the image stream namespace for an immutable Decision Server from KJAR services

If you created image streams in a namespace that is not **openshift**, you must configure the namespace in the template.

If all image streams were already available in your Red Hat OpenShift Container Platform environment, you can skip this procedure.

Prerequisites

- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

If you installed an image streams file according to instructions in Section 2.1, "Ensuring the availability of image streams and the image registry", set the **ImageStream Namespace** (**IMAGE_STREAM_NAMESPACE**) parameter to the name of your OpenShift project.

3.2.4. Configuring information about a Business Central instance for an immutable Decision Server from KJAR services

If you want to enable a connection from a Business Central instance in the same namespace to the Decision Server, you must configure information about the Business Central instance.

Prerequisites

- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

1. Set the following parameters:
   - **KIE Admin User** (**KIE_ADMIN_USER**) and **KIE Admin Password** (**KIE_ADMIN_PWD**): The user name and password for the administrative user. These values must be the same as the **KIE_ADMIN_USER** and **KIE_ADMIN_PWD** settings for the Business Central. If the Business Central uses RH-SSO or LDAP authentication, these values must be a user name and password configured in the authentication system with an administrator role for the Business Central.
   - **Name of the Business Central service**(**DECISION_CENTRAL_SERVICE**): The OpenShift service name for the Business Central.
2. Ensure that the following settings are set to the same value as the same settings for the Business Central:

- **Maven repository URL** (**MAVEN_REPO_URL**): A URL for the external Maven repository froh which services must be deployed.
- **Maven repository username** (**MAVEN_REPO_USERNAME**): The username for the Maven repository.
- **Maven repository password** (**MAVEN_REPO_PASSWORD**): The password for the Maven repository.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

3.2.5. Configuring access to a Maven mirror in an environment without a connection to the public Internet for an immutable Decision Server from KJAR services

When configuring the template to deploy an immutable Decision Server from KJAR services, if your OpenShift environment does not have a connection to the public Internet, you must configure access to a Maven mirror that you set up according to Section 2.4, "Preparing a Maven mirror repository for offline use".

Prerequisites

- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

To configure access to the Maven mirror, set the following parameters:

- **Maven mirror URL** (**MAVEN_MIRROR_URL**): The URL for the Maven mirror repository that you set up in Section 2.4, "Preparing a Maven mirror repository for offline use" . This URL must be accessible from a pod in your OpenShift environment.
- **Maven mirror of** (**MAVEN_MIRROR_OF**): The value that determines which artifacts are to be retrieved from the mirror. For instructions about setting the **mirrorOf** value, see Mirror Settings in the Apache Maven documentation. The default value is **external:***. With this value, Maven retrieves every required artifact from the mirror and does not query any other repositories.
  - If you configure an external Maven repository (**MAVEN_REPO_URL**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror, for example, **external:*,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.
  - If you configure a built-in Business Central Maven repository (**BUSINESS_CENTRAL_MAVEN_SERVICE**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror: **external:*,!repo-rhdmcentr**.
  - If you configure both repositories, change **MAVEN_MIRROR_OF** to exclude the artifacts in both repositories from the mirror: **external:*,!repo-rhdmcentr,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.

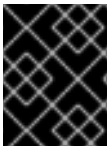
Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

3.2.6. Setting parameters for RH-SSO authentication for an immutable Decision Server from KJAR services

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy an immutable Decision Server from KJAR services.



IMPORTANT

Do not configure LDAP authentication and RH-SSO authentication in the same deployment.

Prerequisites

- A realm for Red Hat Decision Manager is created in the RH-SSO authentication system.
- User names and passwords for Red Hat Decision Manager are created in the RH-SSO authentication system. For a list of the available roles, see Chapter 4, *Red Hat Decision Manager roles and users*. In order to set the parameters for the environment, an administrative user with the **kie-server,rest-all,admin** roles is required. The default user name for this user is **adminUser**. This user can administer and use the environment.
- Clients are created in the RH-SSO authentication system for all components of the Red Hat Decision Manager environment that you are deploying. The client setup contains the URLs for the components. You can review and edit the URLs after deploying the environment. Alternatively, the Red Hat Decision Manager deployment can create the clients. However, this option provides less detailed control over the environment.
- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

1. Set the **KIE_ADMIN_USER** and **KIE_ADMIN_PASSWORD** parameters of the template to the user name and password of the administrative user that you created in the RH-SSO authentication system.
2. Set the following parameters:
   - **RH-SSO URL** (**SSO_URL**): The URL for RH-SSO.
   - **RH-SSO Realm name** (**SSO_REALM**): The RH-SSO realm for Red Hat Decision Manager.
   - **RH-SSO Disable SSL Certificate Validation** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): Set to **true** if your RH-SSO installation does not use a valid HTTPS certificate.
3. Complete one of the following procedures:
   a. If you created the client for Red Hat Decision Manager within RH-SSO, set the following parameters in the template:
      - **Business Central RH-SSO Client name**(**DECISION_CENTRAL_SSO_CLIENT**): The RH-SSO client name for Business Central.

- **KIE Server RH-SSO Client name** (**KIE_SERVER_SSO_CLIENT**): The RH-SSO client name for Decision Server.
- **KIE Server RH-SSO Client Secret** (**KIE_SERVER_SSO_SECRET**): The secret string that is set in RH-SSO for the client for Decision Server.

b. To create the clients for Red Hat Decision Manager within RH-SSO, set the following parameters in the template:

- **KIE Server RH-SSO Client name** (**KIE_SERVER_SSO_CLIENT**): The name of the client to create in RH-SSO for Decision Server.
- **KIE Server RH-SSO Client Secret** (**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for Decision Server.
- **RH-SSO Realm Admin Username** (**SSO_USERNAME**) and **RH-SSO Realm Admin Password** (**SSO_PASSWORD**): The user name and password for the realm administrator user for the RH-SSO realm for Red Hat Decision Manager. You must provide this user name and password in order to create the required clients.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

After completing the deployment, review the URLs for components of Red Hat Decision Manager in the RH-SSO authentication system to ensure they are correct.

3.2.7. Setting parameters for LDAP authentication for an immutable Decision Server from KJAR services

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an immutable Decision Server from KJAR services.

IMPORTANT

Do not configure LDAP authentication and RH-SSO authentication in the same deployment.

Prerequisites

- You created user names and passwords for Red Hat Decision Manager in the LDAP system. For a list of the available roles, see Chapter 4, *Red Hat Decision Manager roles and users* . As a minimum, in order to set the parameters for the environment, you created the following users:
  - An administrative user with the **kie-server,rest-all,admin** roles. This user can administer and use the environment.
  - A server user with the **kie-server,rest-all,user** roles. This user can make REST API calls to the Decision Server.
- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

1. In the LDAP service, create all user names in the deployment parameters. If you do not set any of the parameters, create users with the default user names. The created users must also be assigned to roles:
   - **KIE_ADMIN_USER**: default user name **adminUser**, roles: **kie-server,rest-all,admin**
   - **KIE_SERVER_USER**: default user name **executionUser**, roles **kie-server,rest-all,guest**
     For the user roles that you can configure in LDAP, see Roles and users.
2. Set the **AUTH_LDAP*** parameters of the template. These parameters correspond to the settings of the **LdapExtended** Login module of Red Hat JBoss EAP. For instructions about using these settings, see LdapExtended login module .
   If the LDAP server does not define all the roles required for your deployment, you can map LDAP groups to Red Hat Decision Manager roles. To enable LDAP role mapping, set the following parameters:
   - **RoleMapping rolesProperties file path** (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): The fully qualified path name of a file that defines role mapping, for example, **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**. You must provide this file and mount it at this path in all applicable deployment configurations; for instructions, see Section 3.3, "(Optional) Providing the LDAP role mapping file" .
   - **RoleMapping replaceRole property** (**AUTH_ROLE_MAPPER_REPLACE_ROLE**): If set to **true**, mapped roles replace the roles defined on the LDAP server; if set to **false**, both mapped roles and roles defined on the LDAP server are set as user application roles. The default setting is **false**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

3.2.8. Enabling Prometheus metric collection for an immutable Decision Server from KJAR services

If you want to configure your Decision Server deployment to use Prometheus to collect and store metrics, enable support for this feature in Decision Server at deployment time.

Prerequisites

- You started the configuration of the template, as described in Section 3.2.1, "Starting configuration of the template for an immutable Decision Server from KJAR services".

Procedure

To enable support for Prometheus metric collection, set the **Prometheus Server Extension Disabled** (**PROMETHEUS_SERVER_EXT_DISABLED**) parameter to **false**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.2.9, "Completing deployment of the template for an immutable Decision Server from KJAR services".

For instructions about configuring Prometheus metrics collection, see *Managing and monitoring Decision Server*.

3.2.9. Completing deployment of the template for an immutable Decision Server from KJAR services

After setting all the required parameters in the OpenShift Web UI or in the command line, complete deployment of the template.

Procedure

Depending on the method that you are using, complete the following steps:

- In the OpenShift Web UI, click **Create**.
  - If the **This will create resources that may have security or project behavior implications** message appears, click **Create Anyway**.
- Complete the command line and press Enter.

3.3. (OPTIONAL) PROVIDING THE LDAP ROLE MAPPING FILE

If you configure the **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** parameter, you must provide a file that defines the role mapping. Mount this file on all affected deployment configurations.

Procedure

1. Create the role mapping properties file, for example, **my-role-map**. The file must contain entries in the following format:

   ```
   ldap_role = product_role1, product_role2...
   ```

   For example:

   ```
   admins = kie-server,rest-all,admin
   ```

2. Create an OpenShift configuration map from the file by entering the following command:

   ```
   oc create configmap ldap-role-mapping --from-file=<new_name>=<existing_name>
   ```

   Replace **<new_name>** with the name that the file is to have on the pods (it must be the same as the name specified in the **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** file) and **<existing_name>** with the name of the file that you created. Example:

   ```
   oc create configmap ldap-role-mapping --from-file=rolemapping.properties=my-role-map
   ```

3. Mount the configuration map on every deployment configuration that is configured for role mapping.
   The following deployment configurations can be affected in this environment:
   - ***myapp*-kieserver**: Decision Server

Replace **myapp** with the application name. Sometimes, several Decision Server deployments can be present under different application names.

For every deployment configuration, run the command:

```
oc set volume dc/<deployment_config_name> --add --type configmap --configmap-name
ldap-role-mapping --mount-path=<mapping_dir> --name=ldap-role-mapping
```

Replace **<mapping_dir>** with the directory name (without file name) set in the **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** parameter, for example, **/opt/eap/standalone/configuration/rolemapping** .

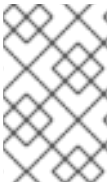
CHAPTER 4. RED HAT DECISION MANAGER ROLES AND USERS

To access Business Central or Decision Server, you must create users and assign them appropriate roles before the servers are started.

The Business Central and Decision Server use Java Authentication and Authorization Service (JAAS) login module to authenticate the users. If both Business Central and Decision Server are running on a single instance, then they share the same JAAS subject and security domain. Therefore, a user, who is authenticated for Business Central can also access Decision Server.

However, if Business Central and Decision Server are running on different instances, then the JAAS login module is triggered for both individually. Therefore, a user, who is authenticated for Business Central, needs to be authenticated separately to access the Decision Server (for example, to view or manage process definitions in Business Central). In case, the user is not authenticated on the Decision Server, then 401 error is logged in the log file, displaying **Invalid credentials to load data from remote server. Contact your system administrator.** message in Business Central.

This section describes available Red Hat Decision Manager user roles.



NOTE

The **admin**, **analyst**, and **rest-all** roles are reserved for Business Central. The **kie-server** role is reserved for Decision Server. For this reason, the available roles can differ depending on whether Business Central, Decision Server, or both are installed.

- **admin**: Users with the **admin** role are the Business Central administrators. They can manage users and create, clone, and manage the repositories. They have full access to make required changes in the application. Users with the **admin** role have access to all areas within Red Hat Decision Manager.
- **analyst**: Users with the **analyst** role have access to all high-level features. They can model projects. However, these users cannot add contributors to spaces or delete spaces in the **Design → Projects** view. Access to the **Deploy → Execution Servers** view, which is intended for administrators, is not available to users with the **analyst** role. However, the **Deploy** button is available to these users when they access the Library perspective.
- **rest-all**: Users with the **rest-all** role can access Business Central REST capabilities.
- **kie-server**: Users with the **kie-server** role can access Decision Server (KIE Server) REST capabilities.

CHAPTER 5. OPENSHIFT TEMPLATE REFERENCE INFORMATION

Red Hat Decision Manager provides the following OpenShift templates. To access the templates, download and extract the **rhdm-7.5.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat customer portal.

- **rhdm75-prod-immutable-kieserver.yaml** provides an immutable Decision Server. Deployment of this template includes a source-to-image (S2I) build for one or several services that are to run on the Decision Server. For details about this template, see Section 5.1, "rhdm75-prod-immutable-kieserver.yaml template".
- **rhdm75-prod-immutable-kieserver-amq.yaml** provides an immutable Decision Server. Deployment of this template includes a source-to-image (S2I) build for one or several services that are to run on the Decision Server. This version of the template includes JMS integration. For details about this template, see Section 5.2, "rhdm75-prod-immutable-kieserver-amq.yaml template".

5.1. RHDM75-PROD-IMMUTABLE-KIESERVER.YAML TEMPLATE

Application template for an immutable KIE server in a production environment, for Red Hat Decision Manager 7.5 - Deprecated

5.1.1. Parameters

Templates allow you to define parameters which take on a value. That value is then substituted wherever the parameter is referenced. References can be defined in any text field in the objects list field. Refer to the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
KIE_ADMIN_USER	**KIE_ADMIN_USER**	KIE administrator username.	adminUser	False
KIE_ADMIN_PWD	**KIE_ADMIN_PWD**	KIE administrator password.	–	False
KIE_SERVER_USER	**KIE_SERVER_USER**	KIE server username. (Sets the org.kie.server.user system property)	executionUser	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_PWD	**KIE_SERVER_PWD**	KIE server password. If this parameter is not set, the password is automatically generated. (Sets the org.kie.server.pwd system property)	–	False
IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStreams for Red Hat Decision Manager images are installed. These ImageStreams are normally installed in the openshift namespace. You should only need to modify this if you installed the ImageStreams in a different namespace/project.	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	The name of the image stream to use for KIE server. Default is "rhdm-kieserver-rhel8".	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	A named pointer to an image in an image stream. Default is "7.5.0".	7.5.0	True
KIE_MBEANS	**KIE_MBEANS**	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	enabled	False

Variable name	Image Environment Variable	Description	Example value	Required
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE server class filtering. (Sets the org.drools.server.filter.classes system property)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	If set to false, the prometheus server extension will be enabled. (Sets the org.kie.prometheus.server.ext.disabled system property)	false	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER_HTTPS_SECRET	–	The name of the secret containing the keystore file.	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret.	keystore.jks	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate.	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate.	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	Allows the KIE server to bypass the authenticated user for task-related operations, for example, queries. (Sets the org.kie.server.bypass.auth.user system property)	false	False
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version |c2(alias2)=g2:a2:v2	rhdm-kieserver-hellorules=org.openshift.quickstarts:rhdm-kieserver-hellorules:1.5.0-SNAPSHOT	True
SOURCE_REPOSITORY_URL	–	Git source URI for application.	https://github.com/jboss-container-images/rhdm-7-openshift-image.git	True
SOURCE_REPOSITORY_REF	–	Git branch/tag reference.	7.5.x	False
CONTEXT_DIR	–	Path within Git project to build; empty for root project directory.	quickstarts/hello-rules/hellorules	False

Variable name	Image Environment Variable	Description	Example value	Required
GITHUB_WEBHOOK_SECRET	–	GitHub trigger secret.	–	True
GENERIC_WEBHOOK_SECRET	–	Generic build trigger secret.	–	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Maven mirror that KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	–	False
MAVEN_MIRROR_OF	**MAVEN_MIRROR_OF**	Maven mirror configuration for KIE server.	external:*	False
MAVEN_REPO_ID	**EXTERNAL_MAVEN_REPO_ID**	The id to use for the maven repository. If set, it can be excluded from the optionally configured mirror by adding it to MAVEN_MIRROR_OF. For example: external:*,!repo-rhdmcentr,!repo-custom. If MAVEN_MIRROR_URL is set but MAVEN_MIRROR_ID is not set, an id will be generated randomly, but won't be usable in MAVEN_MIRROR_OF.	repo-custom	False
MAVEN_REPO_URL	**EXTERNAL_MAVEN_REPO_URL**	Fully qualified URL to a Maven repository.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	Username to access the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	**EXTERNAL_MAVEN_REPO_PASSWORD**	Password to access the Maven repository, if required.	–	False
DECISION_CENTRAL_SERVICE	**WORKBENCH_SERVICE_NAME**	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	myapp-rhdmcentr	False
DECISION_CENTRAL_MAVEN_USERNAME	**RHDMCENTR_MAVEN_REPO_USERNAME**	Username to access the Maven service hosted by Decision Central inside EAP.	mavenUser	False
DECISION_CENTRAL_MAVEN_PASSWORD	**RHDMCENTR_MAVEN_REPO_PASSWORD**	Password to access the Maven service hosted by Decision Central inside EAP.	maven1!	False
ARTIFACT_DIR	–	List of directories from which archives will be copied into the deployment folder. If unspecified, all archives in /target will be copied.	–	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_MGMT_DISABLED	**KIE_SERVER_MGMT_DISABLED**	Disable management api and don't allow KIE containers to be deployed/undeployed or started/stopped. Sets the property org.kie.server.mgmt.api.disabled to true and org.kie.server.startup.strategy to LocalContainersStartupStrategy.	true	True
SSO_URL	**SSO_URL**	RH-SSO URL	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name.	–	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server RH-SSO Client name.	–	False
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	RH-SSO Realm Admin Username used to create the Client if it doesn't exist.	–	False
SSO_PASSWORD	**SSO_PASSWORD**	RH-SSO Realm Admin Password used to create the Client.	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO Disable SSL Certificate Validation.	false	False

Variable name	Image Environment Variable	Description	Example value	Required
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	RH-SSO Principal Attribute to use as username.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP Endpoint to connect for authentication.	ldap://myldap.example.com	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=exmample,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	LDAP Credentials used for authentication.	Password	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	LDAP Base DN of the top-level context to begin the user search.	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	The search scope to use.	**SUBTREE_SCOPE**	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	The timeout in milliseconds for user or role searches.	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	distinguishedName	False
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with usernameBeginString and usernameEndString.	true	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	Name of the attribute containing the user roles.	memberOf	False
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	ou=groups,ou=example,ou=com	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	(memberOf={1})	False
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	A role included for all authenticated users.	user	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributeIsDN property is set to true, this property is used to find the role object's name attribute.	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	A flag indicating if the DN returned by a query contains the roleNameAttribute ID. If set to true, the DN is checked for the roleNameAttribute ID. If set to false, the DN is not checked for the roleNameAttribute ID. This flag can improve the performance of LDAP queries.	false	False
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttribute Id attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	false	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	–	False
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	When present, the RoleMapping Login Module will be configured to use the provided file. This parameter defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	–	False

5.1.2. Objects

The CLI supports various object types. A list of these object types as well as their abbreviations can be found in the Openshift documentation.

5.1.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. Refer to the container-engine documentation for more information.

Service	Port	Name	Description
${APPLICATION_NAME}-kieserver	8080	http	All the KIE server web server's ports.
	8443	https	
${APPLICATION_NAME}-kieserver-ping	8888	ping	The JGroups ping port for clustering.

5.1.2.2. Routes

A route is a way to expose a service by giving it an externally-reachable hostname such as **www.example.com**. A defined route and the endpoints identified by its service can be consumed by a router to provide named connectivity from external clients to your applications. Each route consists of a route name, service selector, and (optionally) security configuration. Refer to the Openshift documentation for more information.

Service	Security	Hostname
insecure-${APPLICATION_NAME}-kieserver-http	none	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS passthrough	**${KIE_SERVER_HOSTNAME_HTTPS}**

5.1.2.3. Build Configurations

A **buildConfig** describes a single build definition and a set of triggers for when a new build should be created. A **buildConfig** is a REST object, which can be used in a POST to the API server to create a new instance. Refer to the Openshift documentation for more information.

S2I image	link	Build output	BuildTriggers and Settings
rhdm-kieserver-rhel8:7.5.0	**rhdm-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub, Generic, ImageChange, ConfigChange

5.1.2.4. Deployment Configurations

A deployment in OpenShift is a replication controller based on a user defined template called a deployment configuration. Deployments are created manually or in response to triggered events. Refer to the Openshift documentation for more information.

5.1.2.4.1. Triggers

A trigger drives the creation of new deployments in response to events, both inside and outside OpenShift. Refer to the Openshift documentation for more information.

Deployment	Triggers
${APPLICATION_NAME}-kieserver	ImageChange

5.1.2.4.2. Replicas

A replication controller ensures that a specified number of pod "replicas" are running at any one time. If there are too many, the replication controller kills some pods. If there are too few, it starts more. Refer to the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-kieserver	2

5.1.2.4.3. Pod Template

5.1.2.4.3.1. Service Accounts

Service accounts are API objects that exist within each project. They can be created or deleted like any other API object. Refer to the Openshift documentation for more information.

Deployment	Service Account
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

5.1.2.4.3.2. Image

Deployment	Image
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

5.1.2.4.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

5.1.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

5.1.2.4.3.5. Exposed Ports

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**

5.1.2.4.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	**${DECISION_CENTRAL_SERVICE}**
	KIE_ADMIN_USER	KIE administrator username.	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	KIE administrator password.	**${KIE_ADMIN_PWD}**
	KIE_SERVER_MODE	–	**DEVELOPMENT**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering. (Sets the org.drools.server.filter.classes system property)	**${DROOLS_SERVER_FILTER_CLASSES}**

Deployment	Variable name	Description	Example value
	PROMETHEUS_SERVER_EXT_DISABLED	If set to false, the prometheus server extension will be enabled. (Sets the org.kie.prometheus.server.ext.disabled system property)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	Allows the KIE server to bypass the authenticated user for task-related operations, for example, queries. (Sets the org.kie.server.bypass.auth.user system property)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_USER	KIE server username. (Sets the org.kie.server.user system property)	**${KIE_SERVER_USER}**
	KIE_SERVER_PWD	KIE server password. If this parameter is not set, the password is automatically generated. (Sets the org.kie.server.pwd system property)	**${KIE_SERVER_PWD}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**

Deployment	Variable name	Description	Example value
	MAVEN_MIRROR_URL	Maven mirror that KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	Maven mirror configuration for KIE server.	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	**${DECISION_CENTRAL_SERVICE}**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	Username to access the Maven service hosted by Decision Central inside EAP.	**${DECISION_CENTRAL_MAVEN_USERNAME}**
	RHDMCENTR_MAVEN_REPO_PASSWORD	Password to access the Maven service hosted by Decision Central inside EAP.	**${DECISION_CENTRAL_MAVEN_PASSWORD}**

Deployment	Variable name	Description	Example value
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository. If set, it can be excluded from the optionally configured mirror by adding it to MAVEN_MIRROR_OF. For example: external:*,!repo-rhdmcentr,!repo-custom. If MAVEN_MIRROR_URL is set but MAVEN_MIRROR_ID is not set, an id will be generated randomly, but won't be usable in MAVEN_MIRROR_OF.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret.	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate.	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	The password for the keystore and certificate.	**${KIE_SERVER_HTTPS_PASSWORD}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_MGMT_DISABLED	Disable management api and don't allow KIE containers to be deployed/undeployed or started/stopped. Sets the property org.kie.server.mgmt.api.disabled to true and org.kie.server.startup.strategy to LocalContainersStartupStrategy.	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**
	SSO_SECRET	KIE Server RH-SSO Client Secret.	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server RH-SSO Client name.	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn't exist.	**${SSO_USERNAME}**
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client.	**${SSO_PASSWORD}**

Deployment	Variable name	Description	Example value
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication.	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	**${AUTH_LDAP_BASE_CTX_DN}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with usernameBeginString and usernameEndString.	**${AUTH_LDAP_PARSE_USERNAME}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	Name of the attribute containing the user roles.	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	**${AUTH_LDAP_ROLES_CTX_DN}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users.	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributeIsDN property is set to true, this property is used to find the role object's name attribute.	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeId attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

Deployment	Variable name	Description	Example value
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This parameter defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**

5.1.2.4.3.7. Volumes

Deployment	Name	mountPath	Purpose	readOnly
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True

5.1.2.5. External Dependencies

5.1.2.5.1. Secrets

This template requires the following secrets to be installed for the application to run.

kieserver-app-secret

5.2. RHDM75-PROD-IMMUTABLE-KIESERVER-AMQ.YAML TEMPLATE

Application template for an immutable KIE server in a production environment integrated with ActiveMQ, for Red Hat Decision Manager 7.5 - Deprecated

5.2.1. Parameters

Templates allow you to define parameters which take on a value. That value is then substituted wherever the parameter is referenced. References can be defined in any text field in the objects list field. Refer to the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
KIE_ADMIN_USER	**KIE_ADMIN_USER**	KIE administrator username	adminUser	False
KIE_ADMIN_PWD	**KIE_ADMIN_PWD**	KIE administrator password	–	False
KIE_SERVER_USER	**KIE_SERVER_USER**	KIE server username (Sets the org.kie.server.user system property)	executionUser	False
KIE_SERVER_PWD	**KIE_SERVER_PWD**	KIE server password. If this parameter is not set, the password is automatically generated. (Sets the org.kie.server.pwd system property)	–	False
IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStreams for Red Hat Decision Manager images are installed. These ImageStreams are normally installed in the openshift namespace. You should only need to modify this if you installed the ImageStreams in a different namespace/project.	openshift	True

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_IMAGE_STREAM_NAME	–	The name of the image stream to use for KIE server. Default is "rhdm-kieserver-rhel8".	rhdm-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	A named pointer to an image in an image stream. Default is "7.5.0".	7.5.0	True
KIE_MBEANS	**KIE_MBEANS**	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	enabled	False
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	true	False
PROMETHEUS_SERVER_EXT_DISABLED	**PROMETHEUS_SERVER_EXT_DISABLED**	If set to false, the prometheus server extension will be enabled. (Sets the org.kie.prometheus.server.ext.disabled system property)	false	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER_HTTPS_SECRET	–	The name of the secret containing the keystore file	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate	mykeystorepass	False
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	Allows the KIE server to bypass the authenticated user for task-related operations, for example, queries. (Sets the org.kie.server.bypass.auth.user system property)	false	False
KIE_SERVER_CONTAINER_DEPLOYMENT	**KIE_SERVER_CONTAINER_DEPLOYMENT**	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version |c2(alias2)=g2:a2:v2	rhdm-kieserver-hellorules=org.openshift.quickstarts:rhdm-kieserver-hellorules:1.5.0-SNAPSHOT	True

Variable name	Image Environment Variable	Description	Example value	Required
SOURCE_REPOSITORY_URL	–	Git source URI for application	https://github.com/jboss-container-images/rhdm-7-openshift-image.git	True
SOURCE_REPOSITORY_REF	–	Git branch/tag reference	7.5.x	False
CONTEXT_DIR	–	Path within Git project to build; empty for root project directory.	quickstarts/hello-rules/hellorules	False
GITHUB_WEBHOOK_SECRET	–	GitHub trigger secret	–	True
GENERIC_WEBHOOK_SECRET	–	Generic build trigger secret	–	True
MAVEN_MIRROR_URL	–	Maven mirror to use for S2I builds	–	False
MAVEN_REPO_ID	**EXTERNAL_MAVEN_REPO_ID**	The id to use for the maven repository, if set. Default is generated randomly.	my-repo-id	False
MAVEN_REPO_URL	**EXTERNAL_MAVEN_REPO_URL**	Fully qualified URL to a Maven repository.	–	False
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	Username to access the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	**EXTERNAL_MAVEN_REPO_PASSWORD**	Password to access the Maven repository, if required.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_SERVICE	**WORKBENCH_SERVICE_NAME**	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	myapp-rhdmcentr	False
DECISION_CENTRAL_MAVEN_USERNAME	**RHDMCENTR_MAVEN_REPO_USERNAME**	Username to access the Maven service hosted by Decision Central inside EAP.	mavenUser	False
DECISION_CENTRAL_MAVEN_PASSWORD	**RHDMCENTR_MAVEN_REPO_PASSWORD**	Password to access the Maven service hosted by Decision Central inside EAP.	maven1!	False
ARTIFACT_DIR	–	List of directories from which archives will be copied into the deployment folder. If unspecified, all archives in /target will be copied.	–	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit	1Gi	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_MGMT_DISABLED	**KIE_SERVER_MGMT_DISABLED**	Disable management api and don't allow KIE containers to be deployed/undeployed or started/stopped. Sets the property org.kie.server.mgmt.api.disabled to true and org.kie.server.startup.strategy to LocalContainersStartupStrategy.	true	True
KIE_SERVER_JMS_QUEUE_REQUEST	**KIE_SERVER_JMS_QUEUE_REQUEST**	JNDI name of request queue for JMS. The default value is queue/KIE.SERVER.REQUEST	queue/KIE.SERVER.REQUEST	False
KIE_SERVER_JMS_QUEUE_RESPONSE	**KIE_SERVER_JMS_QUEUE_RESPONSE**	JNDI name of response queue for JMS. The default value is queue/KIE.SERVER.RESPONSE	queue/KIE.SERVER.RESPONSE	False
AMQ_USERNAME	**AMQ_USERNAME**	User name for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	–	False
AMQ_PASSWORD	**AMQ_PASSWORD**	Password for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	–	False
AMQ_ROLE	**AMQ_ROLE**	User role for standard broker user.	admin	True

Variable name	Image Environment Variable	Description	Example value	Required
AMQ_QUEUES	**AMQ_QUEUES**	Queue names, separated by commas. These queues will be automatically created when the broker starts. Also, they will be made accessible as JNDI resources in EAP. These are the default queues needed by KIE Server. If using custom Queues, use the same values here as in the KIE_SERVER_JMS_QUEUE_RESPONSE and KIE_SERVER_JMS_QUEUE_REQUEST parameters.	queue/KIE.SERVER.REQUEST,queue/KIE.SERVER.RESPONSE	False
AMQ_GLOBAL_MAX_SIZE	**AMQ_GLOBAL_MAX_SIZE**	Specifies the maximum amount of memory that message data can consume. If no value is specified, half of the system's memory is allocated.	10 gb	False
AMQ_SECRET	–	The name of a secret containing AMQ SSL related files.	broker-app-secret	True
AMQ_TRUSTSTORE	**AMQ_TRUSTSTORE**	The name of the AMQ SSL Trust Store file.	broker.ts	False
AMQ_TRUSTSTORE_PASSWORD	**AMQ_TRUSTSTORE_PASSWORD**	The password for the AMQ Trust Store.	changeit	False

Variable name	Image Environment Variable	Description	Example value	Required
AMQ_KEYSTORE	**AMQ_KEYSTORE**	The name of the AMQ keystore file.	broker.ks	False
AMQ_KEYSTORE_PASSWORD	**AMQ_KEYSTORE_PASSWORD**	The password for the AMQ keystore and certificate.	changeit	False
AMQ_PROTOCOL	**AMQ_PROTOCOL**	Broker protocols to configure, separated by commas. Allowed values are: **openwire**, **amqp**, **stomp** and **mqtt**. Only **openwire** is supported by EAP.	openwire	False
AMQ_BROKER_IMAGESTREAM_NAME	–	AMQ Broker Image Stream Name	amq-broker:7.4	True
AMQ_IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStreams for Red Hat AMQ images are installed. These ImageStreams are normally installed in the openshift namespace. You should only need to modify this if you installed the ImageStreams in a different namespace/project.	openshift	True
SSO_URL	**SSO_URL**	RH-SSO URL	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name	–	False
KIE_SERVER_SSO_CLIENT	**SSO_CLIENT**	KIE Server RH-SSO Client name	–	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_SSO_SECRET	**SSO_SECRET**	KIE Server RH-SSO Client Secret	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	RH-SSO Realm Admin Username used to create the Client if it doesn't exist	–	False
SSO_PASSWORD	**SSO_PASSWORD**	RH-SSO Realm Admin Password used to create the Client	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO Disable SSL Certificate Validation	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	RH-SSO Principal Attribute to use as username.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP Endpoint to connect for authentication	ldap://myldap.example.com	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication	uid=admin,ou=users,ou=exmample,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	LDAP Credentials used for authentication	Password	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	LDAP Base DN of the top-level context to begin the user search.	ou=users,ou=example,ou=com	False
AUTH_LDAP_BASE_FILTER	**AUTH_LDAP_BASE_FILTER**	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	(uid={0})	False
AUTH_LDAP_SEARCH_SCOPE	**AUTH_LDAP_SEARCH_SCOPE**	The search scope to use.	**SUBTREE_SCOPE**	False
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	The timeout in milliseconds for user or role searches.	10000	False
AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	**AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE**	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	distinguishedName	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with usernameBeginString and usernameEndString.	true	False
AUTH_LDAP_USERNAME_BEGIN_STRING	**AUTH_LDAP_USERNAME_BEGIN_STRING**	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False
AUTH_LDAP_USERNAME_END_STRING	**AUTH_LDAP_USERNAME_END_STRING**	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False
AUTH_LDAP_ROLE_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_ATTRIBUTE_ID**	Name of the attribute containing the user roles.	memberOf	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLES_CTX_DN	**AUTH_LDAP_ROLES_CTX_DN**	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	ou=groups,ou=example,ou=com	False
AUTH_LDAP_ROLE_FILTER	**AUTH_LDAP_ROLE_FILTER**	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	(memberOf={1})	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_RECURSION	**AUTH_LDAP_ROLE_RECURSION**	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	1	False
AUTH_LDAP_DEFAULT_ROLE	**AUTH_LDAP_DEFAULT_ROLE**	A role included for all authenticated users	user	False
AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	**AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID**	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributeIsDN property is set to true, this property is used to find the role object's name attribute.	name	False
AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	**AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN**	A flag indicating if the DN returned by a query contains the roleNameAttribute ID. If set to true, the DN is checked for the roleNameAttribute ID. If set to false, the DN is not checked for the roleNameAttribute ID. This flag can improve the performance of LDAP queries.	false	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	**AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN**	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeId attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	false	False
AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	**AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK**	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_ROLE_MAPPER_ROLES_PROPERTIES	**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	–	False
AUTH_ROLE_MAPPER_REPLACE_ROLE	**AUTH_ROLE_MAPPER_REPLACE_ROLE**	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	–	False

5.2.2. Objects

The CLI supports various object types. A list of these object types as well as their abbreviations can be found in the Openshift documentation.

5.2.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. Refer to the container-engine documentation for more information.

Service	Port	Name	Description
${APPLICATION_NAME}-kieserver	8080	http	All the KIE server web server's ports.
	8443	https	
${APPLICATION_NAME}-kieserver-ping	8888	ping	The JGroups ping port for clustering.

Service	Port	Name	Description
${APPLICATION_NAME}-amq-jolokia	8161	amq-jolokia-console	The broker's console and Jolokia port.
${APPLICATION_NAME}-amq-amqp	5672	amq-amqp	The broker's AMQP port.
${APPLICATION_NAME}-amq-amqp-ssl	5671	amq-amqp-ssl	The broker's AMQP SSL port.
${APPLICATION_NAME}-amq-mqtt	1883	amq-mqtt	The broker's MQTT port.
${APPLICATION_NAME}-amq-mqtt-ssl	8883	amq-mqtt-ssl	The broker's MQTT SSL port.
${APPLICATION_NAME}-amq-stomp	61613	amq-stomp	The broker's STOMP port.
${APPLICATION_NAME}-amq-stomp-ssl	61612	amq-stomp-ssl	The broker's STOMP SSL port.
${APPLICATION_NAME}-amq-tcp	61616	amq-tcp	The broker's OpenWire port.
${APPLICATION_NAME}-amq-tcp-ssl	61617	amq-tcp-ssl	The broker's OpenWire (SSL) port.

5.2.2.2. Routes

A route is a way to expose a service by giving it an externally-reachable hostname such as **www.example.com**. A defined route and the endpoints identified by its service can be consumed by a router to provide named connectivity from external clients to your applications. Each route consists of a route name, service selector, and (optionally) security configuration. Refer to the Openshift documentation for more information.

Service	Security	Hostname
insecure-${APPLICATION_NAME}-kieserver-http	none	**${KIE_SERVER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-https	TLS passthrough	**${KIE_SERVER_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-amq-jolokia-console	TLS passthrough	<default>

Service	Security	Hostname
${APPLICATION_NAME}-amq-tcp-ssl	TLS passthrough	<default>

5.2.2.3. Build Configurations

A **buildConfig** describes a single build definition and a set of triggers for when a new build should be created. A **buildConfig** is a REST object, which can be used in a POST to the API server to create a new instance. Refer to the Openshift documentation for more information.

S2I image	link	Build output	BuildTriggers and Settings
rhdm-kieserver-rhel8:7.5.0	**rhdm-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub, Generic, ImageChange, ConfigChange

5.2.2.4. Deployment Configurations

A deployment in OpenShift is a replication controller based on a user defined template called a deployment configuration. Deployments are created manually or in response to triggered events. Refer to the Openshift documentation for more information.

5.2.2.4.1. Triggers

A trigger drives the creation of new deployments in response to events, both inside and outside OpenShift. Refer to the Openshift documentation for more information.

Deployment	Triggers
${APPLICATION_NAME}-kieserver	ImageChange
${APPLICATION_NAME}-amq	ImageChange

5.2.2.4.2. Replicas

A replication controller ensures that a specified number of pod "replicas" are running at any one time. If there are too many, the replication controller kills some pods. If there are too few, it starts more. Refer to the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-kieserver	2
${APPLICATION_NAME}-amq	1

5.2.2.4.3. Pod Template

5.2.2.4.3.1. Service Accounts

Service accounts are API objects that exist within each project. They can be created or deleted like any other API object. Refer to the Openshift documentation for more information.

Deployment	Service Account
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**

5.2.2.4.3.2. Image

Deployment	Image
${APPLICATION_NAME}-kieserver	**${APPLICATION_NAME}-kieserver**
${APPLICATION_NAME}-amq	**${AMQ_BROKER_IMAGESTREAM_NAME}**

5.2.2.4.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/readycheck

${APPLICATION_NAME}-amq

> /bin/bash -c /opt/amq/bin/readinessProbe.sh

5.2.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

> Http Get on http://localhost:8080/services/rest/server/healthcheck

5.2.2.4.3.5. Exposed Ports

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-amq	console-jolokia	8161	**TCP**
	amq-amqp	5672	**TCP**
	amqp-ssl	5671	**TCP**
	amq-mqtt	1883	**TCP**
	mqtt-ssl	8883	**TCP**
	amq-stomp	61613	**TCP**
	stomp-ssl	61612	**TCP**
	amq-tcp	61616	**TCP**
	amq-tcp-ssl	61617	**TCP**

5.2.2.4.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	**${DECISION_CENTRAL_SERVICE}**
	KIE_ADMIN_USER	KIE administrator username	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	KIE administrator password	**${KIE_ADMIN_PWD}**
	KIE_SERVER_MODE	–	**DEVELOPMENT**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**

Deployment	Variable name	Description	Example value
	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	If set to false, the prometheus server extension will be enabled. (Sets the org.kie.prometheus.server.ext.disabled system property)	**${PROMETHEUS_SERVER_EXT_DISABLED}**
	KIE_SERVER_BYPASS_AUTH_USER	Allows the KIE server to bypass the authenticated user for task-related operations, for example, queries. (Sets the org.kie.server.bypass.auth.user system property)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	**${KIE_SERVER_USER}**
	KIE_SERVER_PWD	KIE server password. If this parameter is not set, the password is automatically generated. (Sets the org.kie.server.pwd system property)	**${KIE_SERVER_PWD}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**

Deployment	Variable name	Description	Example value
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_SERVICE	The Service name for the optional Decision Central, where it can be reached, to allow service lookups (for example, maven repo usage), if required.	**${DECISION_CENTRAL_SERVICE}**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	Username to access the Maven service hosted by Decision Central inside EAP.	**${DECISION_CENTRAL_MAVEN_USERNAME}**
	RHDMCENTR_MAVEN_REPO_PASSWORD	Password to access the Maven service hosted by Decision Central inside EAP.	**${DECISION_CENTRAL_MAVEN_PASSWORD}**
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_JMS_QUEUE_REQUEST	JNDI name of request queue for JMS. The default value is queue/KIE.SERVER.REQUEST	**${KIE_SERVER_JMS_QUEUE_REQUEST}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_JMS_QUEUE_RESPONSE	JNDI name of response queue for JMS. The default value is queue/KIE.SERVER.RESPONSE	**${KIE_SERVER_JMS_QUEUE_RESPONSE}**
	MQ_SERVICE_PREFIX_MAPPING	–	**${APPLICATION_NAME}-amq7=AMQ**
	AMQ_USERNAME	User name for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	**${AMQ_USERNAME}**
	AMQ_PASSWORD	Password for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	**${AMQ_PASSWORD}**
	AMQ_PROTOCOL	Broker protocols to configure, separated by commas. Allowed values are: **openwire**, **amqp**, **stomp** and **mqtt**. Only **openwire** is supported by EAP.	tcp
	AMQ_QUEUES	Queue names, separated by commas. These queues will be automatically created when the broker starts. Also, they will be made accessible as JNDI resources in EAP. These are the default queues needed by KIE Server. If using custom Queues, use the same values here as in the KIE_SERVER_JMS_QUEUE_RESPONSE and KIE_SERVER_JMS_QUEUE_REQUEST parameters.	**${AMQ_QUEUES}**

Deployment	Variable name	Description	Example value
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	The password for the keystore and certificate	**${KIE_SERVER_HTTPS_PASSWORD}**
	KIE_SERVER_MGMT_DISABLED	Disable management api and don't allow KIE containers to be deployed/undeployed or started/stopped. Sets the property org.kie.server.mgmt.api.disabled to true and org.kie.server.startup.strategy to LocalContainersStartupStrategy.	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name	**${SSO_REALM}**

Deployment	Variable name	Description	Example value
	SSO_SECRET	KIE Server RH-SSO Client Secret	**${KIE_SERVER_SSO_SECRET}**
	SSO_CLIENT	KIE Server RH-SSO Client name	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn't exist	**${SSO_USERNAME}**
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication	**${AUTH_LDAP_BIND_DN}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with usernameBeginString and usernameEndString.	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	Name of the attribute containing the user roles.	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**
	AUTH_LDAP_ROLES_CTX_DN	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	**${AUTH_LDAP_ROLES_CTX_DN}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users	**${AUTH_LDAP_DEFAULT_ROLE}**
	AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID	Name of the attribute within the roleCtxDN context which contains the role name. If the roleAttributeIsDN property is set to true, this property is used to find the role object's name attribute.	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	**${AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN}**
	AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN	Whether or not the roleAttributeID contains the fully-qualified DN of a role object. If false, the role name is taken from the value of the roleNameAttributeId attribute of the context name. Certain directory schemas, such as Microsoft Active Directory, require this attribute to be set to true.	**${AUTH_LDAP_ROLE_ATTRIBUTE_IS_DN}**
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**

Deployment	Variable name	Description	Example value
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-amq	**AMQ_USER**	User name for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	**${AMQ_USERNAME}**
	AMQ_PASSWORD	Password for standard broker user. It is required for connecting to the broker. If left empty, it will be generated.	**${AMQ_PASSWORD}**
	AMQ_ROLE	User role for standard broker user.	**${AMQ_ROLE}**
	AMQ_NAME	–	**${APPLICATION_NAME}-broker**
	AMQ_TRANSPORTS	Broker protocols to configure, separated by commas. Allowed values are: **openwire**, **amqp**, **stomp** and **mqtt**. Only **openwire** is supported by EAP.	**${AMQ_PROTOCOL}**

Deployment	Variable name	Description	Example value
	AMQ_QUEUES	Queue names, separated by commas. These queues will be automatically created when the broker starts. Also, they will be made accessible as JNDI resources in EAP. These are the default queues needed by KIE Server. If using custom Queues, use the same values here as in the KIE_SERVER_JMS_QUEUE_RESPONSE and KIE_SERVER_JMS_QUEUE_REQUEST parameters.	**${AMQ_QUEUES}**
	AMQ_GLOBAL_MAX_SIZE	Specifies the maximum amount of memory that message data can consume. If no value is specified, half of the system's memory is allocated.	**${AMQ_GLOBAL_MAX_SIZE}**
	AMQ_REQUIRE_LOGIN	–	true
	AMQ_ANYCAST_PREFIX	–	–
	AMQ_MULTICAST_PREFIX	–	–
	AMQ_KEYSTORE_TRUSTSTORE_DIR	–	**/etc/amq-secret-volume**
	AMQ_TRUSTSTORE	The name of the AMQ SSL Trust Store file.	**${AMQ_TRUSTSTORE}**
	AMQ_TRUSTSTORE_PASSWORD	The password for the AMQ Trust Store.	**${AMQ_TRUSTSTORE_PASSWORD}**
	AMQ_KEYSTORE	The name of the AMQ keystore file.	**${AMQ_KEYSTORE}**

Deployment	Variable name	Description	Example value
	AMQ_KEYSTORE_PASSWORD	The password for the AMQ keystore and certificate.	**${AMQ_KEYSTORE_PASSWORD}**

5.2.2.4.3.7. Volumes

Deployment	Name	mountPath	Purpose	readOnly
${APPLICATION_NAME}-kieserver	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True
${APPLICATION_NAME}-amq	broker-secret-volume	**/etc/amq-secret-volume**	ssl certs	True

5.2.2.5. External Dependencies

5.2.2.5.1. Secrets

This template requires the following secrets to be installed for the application to run.

kieserver-app-secret broker-app-secret

5.3. OPENSHIFT USAGE QUICK REFERENCE

To deploy, monitor, manage, and undeploy Red Hat Decision Manager templates on Red Hat OpenShift Container Platform, you can use the OpenShift Web console or the **oc** command.

For instructions about using the Web console, see Create and build an image using the Web console .

For detailed instructions about using the **oc** command, see CLI Reference. The following commands are likely to be required:

- To create a project, use the following command:

  ```
  $ oc new-project <project-name>
  ```

  For more information, see Creating a project using the CLI .

- To deploy a template (create an application from a template), use the following command:

  ```
  $ oc new-app -f <template-name> -p <parameter>=<value> -p <parameter>=<value> ...
  ```

  For more information, see Creating an application using the CLI .

- To view a list of the active pods in the project, use the following command:

  ```
  $ oc get pods
  ```

- To view the current status of a pod, including information whether or not the pod deployment has completed and it is now in a running state, use the following command:

```
$ oc describe pod <pod-name>
```

  You can also use the **oc describe** command to view the current status of other objects. For more information, see Application modification operations.

- To view the logs for a pod, use the following command:

```
$ oc logs <pod-name>
```

- To view deployment logs, look up a **DeploymentConfig** name in the template reference and enter the following command:

```
$ oc logs -f dc/<deployment-config-name>
```

  For more information, see Viewing deployment logs.

- To view build logs, look up a **BuildConfig** name in the template reference and enter the command:

```
$ oc logs -f bc/<build-config-name>
```

  For more information, see Accessing build logs.

- To scale a pod in the application, look up a **DeploymentConfig** name in the template reference and enter the command:

```
$ oc scale dc/<deployment-config-name> --replicas=<number>
```

  For more information, see Manual scaling.

- To undeploy the application, you can delete the project by using the command:

```
$ oc delete project <project-name>
```

  Alternatively, you can use the **oc delete** command to remove any part of the application, such as a pod or replication controller. For details, see Application modification operations.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Monday, March 01, 2021.