Red Hat Decision Manager 7.9

Deploying Red Hat Decision Manager on Red Hat OpenShift Container Platform

Last Updated: 2022-04-29

Legal Notice

Abstract

This document describes how to deploy a variety of Red Hat Decision Manager environments on Red Hat OpenShift Container Platform, such as an authoring environment, a managed server environment, an immutable server environment, and other supported environment options.

Table of Contents

PREFACE

As a developer or system administrator, you can deploy a variety of Red Hat Decision Manager environments on Red Hat OpenShift Container Platform, such as an authoring environment, a managed server environment, an immutable server environment, and other supported environment options.

MAKING OPEN SOURCE MORE INCLUSIVE

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

PART I. DEPLOYING A RED HAT DECISION MANAGER ENVIRONMENT ON RED HAT OPENSHIFT CONTAINER PLATFORM USING OPERATORS

As a system engineer, you can deploy a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform version 4 to provide an infrastructure to develop or execute services and other business assets. You can use OpenShift Operators to deploy the environment defined in a structured YAML file and to maintain and modify this environment as necessary.

Prerequisites

- A Red Hat OpenShift Container Platform version 4 environment is available. For the exact versions of Red Hat OpenShift Container Platform that the current release supports, see Red Hat Process Automation Manager 7 Supported Configurations.
- The OpenShift project for the deployment is created.
- You are logged into the project using the OpenShift web console.
- The following resources are available on the OpenShift cluster. Depending on the application load, higher resource allocation might be necessary for acceptable performance.
  - For an authoring environment, 4 gigabytes of memory and 2 virtual CPU cores for the Business Central pod. In a high-availability deployment, these resources are required for each replica and two replicas are created by default.
  - 2 gigabytes of memory and 1 virtual CPU core for each replica of each KIE Server pod.
  - In a high-availability authoring deployment, additional resources according to the configured defaults are required for the Red Hat AMQ, and Red Hat Data Grid pods.
- Dynamic persistent volume (PV) provisioning is enabled. Alternatively, if dynamic PV provisioning is not enabled, enough persistent volumes must be available. By default, the deployed components require the following PV sizes:
  - By default, Business Central requires one 1Gi PV. You can change the PV size for Business Central persistent storage.
- If you intend to deploy a high-availability authoring environment, your OpenShift environment supports persistent volumes with **ReadWriteMany** mode. If your environment does not support this mode, you can use NFS to provision the volumes. For information about access mode support in OpenShift public and dedicated clouds, see Access Modes in Red Hat OpenShift Container Platform documentation.

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:

- KIE Server, also known as *Execution 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.
  In some templates, you can scale up a KIE 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 KIE 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 KIE 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 KIE 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.

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

1.1. ARCHITECTURE OF AN AUTHORING ENVIRONMENT

In Red Hat Decision Manager, the Business Central component provides a web-based interactive user interface for authoring services. The KIE Server component runs the services.

You can also use Business Central to deploy services onto a KIE Server. You can use several KIE Servers to run different services and control the servers from the same Business Central.

Single authoring environment

In a single authoring environment, only one instance of Business Central is running. Multiple users can access its web interface at the same time, however the performance can be limited and there is no failover capability.

Business Central includes a built-in Maven repository that stores the built versions of the services that you develop (KJAR files/artifacts). You can use your continuous integration and continuous deployment (CICD) tools to retrieve these artifacts from the repository and move them as necessary.

Business Central saves the source code in a built-in Git repository, stored in the **.niogit** directory. It uses a built-in indexing mechanism to index the assets in your services.

Business Central uses persistent storage for the Maven repository and for the Git repository.

A single authoring environment, by default, includes one KIE Server.

A single authoring environment can use the *controller strategy*. Business Central includes the *Controller*, a component that can manage KIE Servers. When you configure a KIE Server to connect to Business Central, the KIE Server uses a REST API to connect to the Controller. This connection opens a persistent WebSocket. In an OpenShift deployment that uses the controller strategy, each KIE Server is initially configured to connect to the Business Central Controller.

When you use the Business Central user interface to deploy or manage a service on the KIE Server, the KIE Server receives the request through the Controller connection WebSocket. To deploy a service, the KIE Server requests the necessary artifact from the Maven repository that is a part of Business Central.

Client applications use a REST API to use services that run on the KIE Server.

Figure 1.1. Architecture diagram for a single authoring environment

Clustering KIE Servers and using multiple KIE Servers

You can scale a KIE Server pod to run a clustered KIE Server environment.

In a clustered deployment, several instances of the KIE Server run the same services. These servers can connect to the Business Central Controller using the same server ID, so they can receive the same requests from the controller. Red Hat OpenShift Container Platform provides load-balancing between the servers. The services that run on a clustered KIE Server must be stateless, because requests from the same client might be processed by different instances.

You can also deploy several independent KIE Servers to run different services. In this case, the servers connect to the Business Central Controller with different server ID values. You can use the Business Central UI to deploy services to each of the servers.

Smart Router

The optional Smart Router component provides a layer between client applications and KIE Servers. It can be useful if you are using several independent KIE Servers.

The client application can use services running on different KIE Servers, but always connects to the Smart Router. The Smart Router automatically passes the request to the KIE Servers that runs the required service. The Smart Router also enables management of service versions and provides an additional load-balancing layer.

High-availability authoring environment

In a high-availability (HA) authoring environment, the Business Central pod is scaled, so several instances of Business Central are running. Red Hat OpenShift Container Platform provides load balancing for user requests. This environment provides optimal performance for multiple users and supports failover.

Each instance of Business Central includes the Maven repository for the built artifacts and uses the **.niogit** Git repository for source code. The instances use shared persistent storage for the repositories. A persistent volume with **ReadWriteMany** access is required for this storage.

An instance of Red Hat DataGrid provides indexing of all projects and assets developed in Business Central.

An instance of Red Hat AMQ propagates Java CDI messages between all instances of Business Central. For example, when a new project is created or when an asset is locked or modified on one of the instances, this information is immediately reflected in all other instances.

The controller strategy is not suitable for clustered deployment. In an OpenShift deployment, a high-availability Business Central must manage KIE Servers using the *OpenShift startup strategy*.

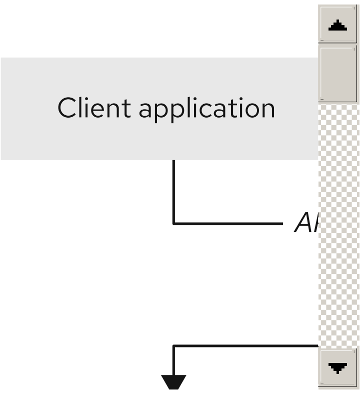
Each KIE Server deployment (which can be scaled) creates a ConfigMap that reflects its current state. The Business Central discovers all KIE Servers by reading their ConfigMaps.

When the user requests a change in KIE Server configuration (for example, deploys or undeploys a service), Business Central initiates a connection to the KIE Server and sends a REST API request. The KIE Server changes the ConfigMap to reflect the new configuration state and then triggers its own redeployment, so that all instances are redeployed and reflect the new configuration.

You can deploy several independent KIE Servers in your OpenShift environment. Each of the KIE Servers has a separate ConfigMap with the necessary configuration. You can scale each of the KIE Servers separately.

You can include Smart Router in the OpenShift deployment.

Figure 1.2. Architecture diagram for a high-availability authoring environment



CHAPTER 2. PREPARATION FOR DEPLOYING RED HAT DECISION MANAGER IN YOUR OPENSHIFT ENVIRONMENT

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

NOTE

If you are deploying a trial environment, complete the procedure described in Section 2.1, “Ensuring your environment is authenticated to the Red Hat registry” and do not complete any other preparation procedures.

2.1. ENSURING YOUR ENVIRONMENT IS AUTHENTICATED TO THE RED HAT REGISTRY

To deploy Red Hat Decision Manager components of Red Hat OpenShift Container Platform, you must ensure that OpenShift can download the correct images from the Red Hat registry.

OpenShift must be configured to authenticate with the Red Hat registry using your service account user name and password. This configuration is specific for a namespace, and if operators work, the configuration is already completed for the **openshift** namespace.

However, if the image streams for Red Hat Decision Manager are not found in the **openshift** namespace or if the operator is configured to update Red Hat Decision Manager to a new version automatically, the operator needs to download images into the namespace of your project. You must complete the authentication configuration for this namespace.

Procedure

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

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

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

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

5. Run 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.

2.2. CREATING THE SECRETS FOR KIE SERVER

OpenShift uses objects called *secrets* to hold sensitive information such as passwords or keystores. For more information about OpenShift secrets, see What is a secret in the Red Hat OpenShift Container Platform documentation.

In order to provide HTTPS access, KIE Server uses an SSL certificate. The deployment can create a sample secret automatically. However, in production environments you must create an SSL certificate for KIE Server and provide it to your OpenShift environment as a secret.

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 KIE Server.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. 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. CREATING THE SECRETS FOR BUSINESS CENTRAL

In order to provide HTTPS access, Business Central uses an SSL certificate. The deployment can create a sample secret automatically. However, in production environments you must create an SSL certificate for Business Central and provide it to your OpenShift environment as a secret.

Do not use the same certificate and keystore for Business Central and KIE Server.

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 Business Central.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. Use the **oc** command to generate a secret named **decisioncentral-app-secret** from the new keystore file:

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

2.4. CREATING THE SECRETS FOR THE AMQ BROKER CONNECTION

If you want to connect any KIE Server to an AMQ broker and to use SSL for the AMQ broker connection, you must create an SSL certificate for the connection and provide it to your OpenShift environment as a secret.

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 the AMQ broker connection.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. Use the **oc** command to generate a secret named **broker-app-secret** from the new keystore file:

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

2.5. PREPARING GIT HOOKS

In an authoring environment you can use Git hooks to execute custom operations when the source code of a project in Business Central is changed. The typical use of Git hooks is for interaction with an upstream repository.

To enable Git hooks to interact with an upstream repository using SSH authentication, you must also provide a secret key and a known hosts file for authentication with the repository.

Skip this procedure if you do not want to configure Git hooks.

Procedure

1. Create the Git hooks files. For instructions, see the Git hooks reference documentation.

   NOTE

   A **pre-commit** script is not supported in Business Central. Use a **post-commit** script.

2. Create a configuration map (ConfigMap) or persistent volume with the files.

   - If the Git hooks consist of one or several fixed script files, use the **oc** command to create a configuration map. For example:

     ```
     oc create configmap git-hooks --from-file=post-commit=post-commit
     ```

   - If the Git hooks consist of long files or depend on binaries, such as executable or JAR files, use a persistent volume. You must create a persistent volume, create a persistent volume claim and associate the volume with the claim, and transfer files to the volume.
     For instructions about persistent volumes and persistent volume claims, see Storage in the Red Hat OpenShift Container Platform documentation. For instructions about copying files onto a persistent volume, see Transferring files in and out of containers .

3. If the Git hooks scripts must interact with an upstream repository using SSH authentication, prepare a secret with the necessary files:

   a. Prepare the **id_rsa** file with a private key that matches a public key stored in the repository.

   b. Prepare the **known_hosts** file with the correct name, address, and public key for the repository.

   c. Create a secret with the two files using the **oc** command, for example:

     ```
     oc create secret git-hooks-secret --from-file=id_rsa=id_rsa --from-
     file=known_hosts=known_hosts
     ```

     NOTE

     When the deployment uses this secret, it mounts the **id_rsa** and **known_hosts** files into the **/home/jboss/.ssh** directory on Business Central pods.

2.6. PROVISIONING PERSISTENT VOLUMES WITH READWRITEMANY ACCESS MODE USING NFS

If you want to deploy high-availability Business Central, your environment must provision persistent volumes with **ReadWriteMany** access mode. If you want to deploy high-availability Business Central, your environment must provision persistent volumes with **ReadWriteMany** access mode.

If your configuration requires provisioning persistent volumes with **ReadWriteMany** access mode but your environment does not support such provisioning, use NFS to provision the volumes. Otherwise, skip this procedure.

Procedure

Deploy an NFS server and provision the persistent volumes using NFS. For information about provisioning persistent volumes using NFS, see the "Persistent storage using NFS" section of the OpenShift Container Platform Storage guide.

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

If you are planning to create immutable KIE servers using the source-to-image (S2I) process, you must provide the source code for your services in a Git repository. 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.

Skip this procedure if you are not planning to use the S2I process or if you are not using Business Central for authoring services.

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.

2.8. PREPARING FOR DEPLOYMENT IN A RESTRICTED NETWORK

You can deploy Red Hat Decision Manager in a restricted network that is not connected to the public Internet. For instructions about operator deployment in a restricted network, see Using Operator Lifecycle Manager on restricted networks in Red Hat OpenShift Container Platform documentation.

IMPORTANT

In Red Hat Decision Manager 7.9, deployment on restricted networks is for Technology Preview only. For more information on Red Hat Technology Preview features, see Technology Preview Features Scope .

In order to use a deployment that does not have outgoing access to the public Internet, you must also prepare a Maven repository with a mirror of all the necessary artifacts. For instructions about creating this repository, see Section 2.9, “Preparing a Maven mirror repository for offline use” .

2.9. 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. Configure a Maven release repository to which you have write access. The repository must allow read access without authentication and your OpenShift environment must have network 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 in the Red Hat OpenShift Container Platform 3.11 documentation. The documented procedure is applicable to Red Hat OpenShift Container Platform version 4.

   Use this repository as a mirror to host the publicly available Maven artifacts. You can also provide your own services in this repository in order to deploy these services on immutable servers.

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

   a. Click **Red Hat Process Automation Manager 7.9.1 Offliner Content List**to download the **rhdm-7.9.1-offliner.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

   b. Extract the contents of the **rhdm-7.9.1-offliner.zip** file into any directory.

   c. Change to the directory and enter the following command:

   ```
   ./offline-repo-builder.sh offliner.txt
   ```

   This command creates a **repository** subdirectory and downloads the necessary artifacts into this subdirectory.

   If a message reports that some downloads have failed, run the same command again. If downloads fail again, contact Red Hat support.

   d. Upload all artifacts from the **repository** subdirectory to the Maven mirror repository that you prepared. You can use the Maven Repository Provisioner utility, available from the Maven repository tools Git repository, 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, available from the Maven repository tools Git repository, to upload the artifacts.

CHAPTER 3. DEPLOYMENT AND MANAGEMENT OF A RED HAT DECISION MANAGER ENVIRONMENT USING OPENSHIFT OPERATORS

To deploy a Red Hat Decision Manager environment, the OpenShift operator uses a YAML source that describes the environment. Red Hat Decision Manager provides an installer that you can use to form the YAML source and deploy the environment.

When the Business Automation operator deploys the environment, it creates a YAML description of the environment, and then ensures that the environment is consistent with the description at all times. You can edit the description to modify the environment.

You can remove the environment by deleting the operator application in Red Hat OpenShift Container Platform.

NOTE

When you remove an environment with a high-availability Business Central, the operator does not delete Persistent Volume Claims that were created as part of the JBoss Datagrid and JBoss AMQ StatefulSet creation. This behaviour is a part of Kubernetes design, as deletion of the Persistent Volume Claims could cause data loss. For more information about handling persistent volumes during deletion of a StatefulSet, see the Kubernetes documentation.

If you create a new environment using the same namespace and the same application name, the environment reuses the persistent volumes for increased performance.

To ensure that new deployments do not use any old data, you can delete the Persistent Volume Claims manually.

3.1. SUBSCRIBING TO THE BUSINESS AUTOMATION OPERATOR

To be able to deploy Red Hat Decision Manager using operators, you must subscribe to the Business Automation operator in OpenShift.

Procedure

1. Enter your project in the OpenShift Web cluster console.
2. In the OpenShift Web console navigation panel, select **Catalog → OperatorHub** or **Operators → OperatorHub**.
3. Search for **Business Automation**, select it and click **Install**.
4. On the **Create Operator Subscription** page, select your target namespace and approval strategy.
   Optional: Set **Approval strategy** to **Automatic** to enable automatic operator updates. An operator update does not immediately update the product, but is required before you update the product. Configure automatic or manual product updates using the settings in every particular product deployment.
5. Click **Subscribe** to create a subscription.

3.2. DEPLOYING A RED HAT DECISION MANAGER ENVIRONMENT USING THE OPERATOR

After you subscribe to the Business Automation operator, you can use the installer wizard to configure and deploy a Red Hat Decision Manager environment.

IMPORTANT

In Red Hat Decision Manager 7.9, the operator installer wizard is for Technology Preview only. For more information on Red Hat Technology Preview features, see Technology Preview Features Support Scope.

3.2.1. Starting the deployment of a Red Hat Decision Manager environment using the Business Automation operator

To start deploying a Red Hat Decision Manager environment using the Business Automation operator, access the installer wizard. The installer wizard is deployed when you subscribe to the operator.

Prerequisites

- You subscribed to the Business Automation operator. For instructions about subscribing to the operator, see Section 3.1, “Subscribing to the Business Automation operator” .

Procedure

1. In the Red Hat OpenShift Container Platform web cluster console menu, select **Catalog → Installed operators** or **Operators → Installed operators**.
2. Click the name of the operator that contains **businessautomation**. Information about this operator is displayed.
3. Click the **Installer** link located on the right side of the window.
4. If prompted, log in with your OpenShift credentials.

Result

The **Installation** tab of the wizard is displayed.

3.2.2. Setting the basic configuration of the environment

After you start to deploy a Red Hat Decision Manager environment using the Business Automation operator, you must select the type of the environment and set other basic configuration.

Prerequisites

- You started to deploy a Red Hat Decision Manager environment using the Business Automation operator and accessed the installer wizard according to the instructions in Section 3.2.1, “Starting the deployment of a Red Hat Decision Manager environment using the Business Automation operator”.

Procedure

1. In the **Application Name** field, enter a name for the OpenShift application. This name is used in the default URLs for all components.

2. In the **Environment** list, select the type of environment. This type determines the default configuration; you can modify this configuration as necessary. The following types are available for Red Hat Decision Manager:

   - **rhdm-trial**: A trial environment that you can set up quickly and use to evaluate or demonstrate developing and running assets. Includes Business Central and a KIE Server. This environment does not use any persistent storage, and any work you do in the environment is not saved.

   - **rhdm-authoring**: An environment for creating and modifying services using Business Central. It consists of pods that provide Business Central for the authoring work and a KIE Server for test execution of the services. You can also use this environment to run services for staging and production purposes. You can add KIE Servers to the environment and they are managed by the same Business Central.

   - **rhdm-authoring-ha**: An environment for creating and modifying services using Business Central. It consists of pods that provide Business Central for the authoring work and a KIE Server for test execution of the services. This version of the authoring environment supports scaling the Business Central pod to ensure high availability.

     IMPORTANT

     In Red Hat Decision Manager 7.9, high-availability Business Central functionality is for Technology Preview only. For more information about Red Hat Technology Preview features, see Technology Preview Features Support Scope.

   - **rhdm-production-immutable**: An alternate environment for running existing services for staging and production purposes. You can configure one or more KIE Server pods that build services from source or pull them from a Maven repository. You can then replicate each pod as necessary.
     You cannot remove any service from the pod or add any new service to the pod. If you want to use another version of a service or to modify the configuration in any other way, deploy a new server image to replace the old one. You can use any container-based integration workflows to manage the pods.

     When configuring this environment, in the **KIE Servers** tab you must customize the KIE Server and either click the **Set immutable server configuration** button or set the **KIE_SERVER_CONTAINER_DEPLOYMENT** environment variable. For instructions about configuring the KIE Server, see Section 3.2.5, "Setting custom KIE Server configuration of the environment".

3. If you want to enable automatic upgrades to new versions, select the **Enable Upgrades** box. If this box is selected, when a new patch version of Red Hat Decision Manager 7.9 becomes available, the operator automatically upgrades your deployment to this version. All services are preserved and normally remain available throughout the upgrade process.
   If you also want to enable the same automatic upgrade process when a new minor version of Red Hat Decision Manager 7.x becomes available, select the **Include minor version upgrades** box.

> NOTE
>
> Disable automatic updates if you want to use a custom image for any component of Red Hat Decision Manager.

4. Optional: If you want to use image tags for downloading images, select the **Use Image Tags** box. This setting is useful if you use a custom registry or if you are directed by Red Hat support.

5. If you want to use a custom image registry, under **Custom registry**, enter the URL of the registry in the **Image registry** field. If this registry does not have a properly signed and recognized SSL certificate, select the **Insecure** box.

> NOTE
>
> To use particular images from the custom registry, set the image context, name, and tag in the **Console** and **KIE Server** tabs.

6. Under **Admin user**, enter the user name and password for the administrative user for Red Hat Decision Manager in the **Username** and **Password** fields.

> IMPORTANT
>
> If you use RH-SSO or LDAP authentication, the same user must be configured in your authentication system with the **kie-server,rest-all,admin** roles for Red Hat Decision Manager.

Next steps

If you want to deploy the environment with the default configuration, click **Finish** and then click **Deploy** to deploy the environment. Otherwise, continue to set other configuration parameters.

3.2.3. Setting the security configuration of the environment

After you set the basic configuration of a Red Hat Decision Manager environment using the Business Automation operator, you can optionally configure authentication (security) settings for the environment.

Prerequisites

- You completed basic configuration of a Red Hat Decision Manager environment using the Business Automation operator in the installer wizard according to the instructions in Section 3.2.2, “Setting the basic configuration of the environment” .

- If you want to use RH-SSO or LDAP for authentication, you created users with the correct roles in your authentication system. You must create at least one administrative user (for example, **adminUser**) with the **kie-server,rest-all,admin** roles. This user must have the user name and password that you configured on the **Installation** tab.

- If you want to use RH-SSO authentication, you created the clients in your RH-SSO system for all components of your environment, specifying the correct URLs. This action ensures maximum control. Alternatively, the deployment can create the clients.

Procedure

1. If the **Installation** tab is open, click **Next** to view the **Security** tab.

2. In the **Authentication mode** list, select one of the following modes:

   - **Internal**: You configure the initial administration user when deploying the environment. The user can use Business Central to set up other users as necessary.

   - **RH-SSO**: Red Hat Decision Manager uses Red Hat Single Sign-On for authentication.

   - **LDAP**: Red Hat Decision Manager uses LDAP for authentication

3. Complete the security configuration based on the **Authentication mode** that you selected.
   If you selected **RH-SSO**, configure RH-SSO authentication:

   a. In the **RH-SSO URL** field, enter the RH-SSO URL.

   b. In the **Realm** field, enter the RH-SSO realm name.

   c. If you did not create RH-SSO clients for components of your environment enter the credentials of an administrative user for your RH-SSO system in the **SSO admin user** and **SSO admin password** fields.

   d. If your RH-SSO system does not have a proper signed SSL certificate, select the **Disable SSL cert validation** box.

   e. If you want to change the RH-SSO principal attribute used for the user name, in the **Principal attribute** field enter the name of the new attribute.

   If you selected **LDAP**, configure LDAP authentication:

   a. In the **LDAP URL** field, enter the LDAP URL.

   b. Configure LDAP parameters that correspond to the settings of the LdapExtended Login module of Red Hat JBoss EAP. For instructions about using these settings, see LdapExtended Login Module.

      NOTE

      If you want to enable LDAP failover, you can set two or more LDAP server addresses in the **AUTH_LDAP_URL** parameter, separated by a space.

4. If you selected **RH-SSO** or **LDAP**, if your RH-SSO or LDAP system does not define all the roles required for your deployment, you can map authentication system roles to Red Hat Decision Manager roles.
   To enable role mapping, you must provide a role mapping configuration file in an OpenShift configuration map or secret object in the project namespace. The file must contain entries in the following format:

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

   For example:

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

   To enable the use of this file, make the following changes:

a. Under **RoleMapper**, in the **Roles properties file** field, enter the fully qualified path name of the role mapping configuration file, for example, **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**.

b. If you want to replace roles defined in the authentication system with roles that you define in the mapping file, select the **Replace roles** box. Otherwise, both the roles defined in RH-SSO or LDAP and the roles defined in the configuration file are available.

c. In the fields under **RoleMapper Configuration object**, select the **Kind** of the object that provides the file (**ConfigMap** or **Secret**) and enter the **Name** of the object. This object is automatically mounted on Business Central and KIE Server pods in the path that you specified for the role mapping configuration file.

5. Configure other passwords, if necessary:

   - **AMQ password** and **AMQ cluster password** are passwords for interaction with ActiveMQ using the JMS API.

   - **Keystore password** is the password for the keystore files used in secrets for HTTPS communication. Set this password if you created secrets according to instructions in Section 2.2, "Creating the secrets for KIE Server" or Section 2.3, "Creating the secrets for Business Central".

   - **Database password** is the password for database server pods that are a part of the environments.

Next steps

If you want to deploy the environment with the default configuration of all components, click **Finish** and then click **Deploy** to deploy the environment. Otherwise, continue to set configuration parameters for Business Central and KIE Servers.

3.2.4. Setting the Business Central configuration of the environment

After you set the basic and security configuration of a Red Hat Decision Manager environment using the Business Automation operator, you can optionally configure settings for the Business Central component of the environment.

All environment types except **rhdm-production-immutable** include this component.

Do not change these settings for the **rhdm-production-immutable** environment, as this environment does not include Business Central or Business Central Monitoring.

Prerequisites

- You completed basic configuration of a Red Hat Decision Manager environment using the Business Automation operator in the installer wizard according to the instructions in Section 3.2.2, "Setting the basic configuration of the environment" .

- If you want to use RH-SSO or LDAP for authentication, you completed security configuration according to the instructions in Section 3.2.3, "Setting the security configuration of the environment".

Procedure

1. If the **Installation** or **Security** tab is open, click **Next** until you view the **Console** tab.

2. If you created the secret for Business Central according to the instructions in Section 2.3, "Creating the secrets for Business Central", enter the name of the secret in the **Keystore secret** field.

3. Optional: If you want to use a custom image for the Business Central deployment, complete the following additional steps:

   a. Set the custom registry in the **Installation** tab. If you do not set the custom registry, the installation uses the default Red Hat registry. For more information about setting the custom registry value, see Section 3.2.2, "Setting the basic configuration of the environment".

   b. In the **Console** tab, set the following fields:

      - **Image context**: The context of the image in the registry.

      - **Image**: The name of the image.

      - **Image tag**: The tag of the image. If you do not set this field, the installation uses the **latest** tag.
        For example, if the full address of the image is **registry.example.com/mycontext/mycentral:1.0-SNAPSHOT**, set the custom registry to **registry.example.com**, the **Image context** field to **mycontext**, the **Image** field to **mycentral**, and the **Image tag** field to **1.0-SNAPSHOT**.

4. Optional: Configure Git hooks.
   In an authoring environment, you can use Git hooks to facilitate interaction between the internal Git repository of Business Central and an external Git repository. If you want to use Git hooks, you must prepare a Git hooks directory in an OpenShift configuration map, secret, or persistent volume claim object in the project namespace. You can also prepare a secret with the SSH key and known hosts files for Git SSH authentication. For instructions about preparing Git hooks, see Section 2.5, "Preparing Git hooks".

   To use a Git hooks directory, make the following changes:

   a. Under **GitHooks**, in the **Mount path** field, enter a fully qualified path for the directory, for example, **/opt/kie/data/git/hooks**.

   b. In the fields under **GitHooks Configuration object**, select the **Kind** of the object that provides the file (**ConfigMap**, **Secret**, or **PersistentVolumeClaim**) and enter the **Name** of the object. This object is automatically mounted on the Business Central pods in the path that you specified for the Git hooks directory.

   c. Optionally, in the **SSH secret** field enter the name of the secret with the SSH key and known hosts files.

5. Optionally, enter the number of replicas for Business Central or Business Central monitoring in the **Replicas** field. Do not change this number in a **rhdm-authoring** environment.

6. Optionally, enter requested and maximum CPU and memory limits in the fields under **Resource quotas**.

7. If you want to customize the configuration of the Java virtual machine on the Business Central pods, select the **Enable JVM configuration** box and then enter information in any of the fields under **Enable JVM configuration**. All fields are optional. For the JVM parameters that you can configure, see Section 3.4, "JVM configuration parameters".

8. If you selected RH-SSO authentication, configure RH-SSO for Business Central:

    a. Enter the client name in the **Client name** field and the client secret in the **Client secret** field. If a client with this name does not exist, the deployment attempts to create a new client with this name and secret.

    b. If the deployment is to create a new client, enter the HTTP and HTTPS URLs that will be used for accessing the KIE Server into the **SSO HTTP URL** and **SSO HTTPS URL** fields. This information is recorded in the client.

9. Optionally, depending on your needs, set environment variables. To set an environment variable, click **Add new Environment variable**, then enter the name and value for the variable in the **Name** and **Value** fields.

    - If you want to use an external Maven repository, set the following variables:

        - **MAVEN_REPO_URL**: The URL for the Maven repository

        - **MAVEN_REPO_ID**: An identifier for the Maven repository, for example, **repo-custom**

        - **MAVEN_REPO_USERNAME**: The user name for the Maven repository

        - **MAVEN_REPO_PASSWORD** The password for the Maven repository

        IMPORTANT

        In an authoring environment, if you want Business Central to push a project into an external Maven repository, you must configure this repository during deployment and also configure exporting to the repository in every project. For information about exporting Business Central projects to an external Maven repository, see *Packaging and deploying a Red Hat Decision Manager project*.

    - If your OpenShift environment does not have a connection to the public Internet, configure access to a Maven mirror that you set up according to Section 2.9, “Preparing a Maven mirror repository for offline use”. Set the following variables:

        - **MAVEN_MIRROR_URL**: The URL for the Maven mirror repository that you set up in Section 2.9, “Preparing a Maven mirror repository for offline use” . This URL must be accessible from a pod in your OpenShift environment.

        - **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 your authoring environment uses a built-in Business Central Maven repository, change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror: **external:*,!repo-rhdmcentr**.

    - In some cases, you might want to persist the Maven repository cache for Business Central. By default, the cache is not persisted, so when you restart or scale a Business Central pod,

all Maven artifacts are downloaded again and all projects within Business Central must be built again. If you enable persistence for the cache, the download is not necessary and startup time can improve in some situations. However, significant additional space on the Business Central persistence volume is required.
To enable persistence for the Maven repository cache, set the **KIE_PERSIST_MAVEN_REPO** environment variable to **true**.

If you set **KIE_PERSIST_MAVEN_REPO** to **true**, you can optionally set a custom path for the cache using the **KIE_M2_REPO_DIR** variable. The default path is **/opt/kie/data/m2**. Files in the **/opt/kie/data** directory tree are persisted.

- In some authoring environments, you might need to ensure that several users can deploy services on the same KIE Server at the same time. By default, after deploying a service onto a KIE Server using Business Central, the user needs to wait for some seconds before more services can be deployed. The **OpenShiftStartupStrategy** setting is enabled by default and causes this limitation. To remove the limitation, you can configure an **rhdm-authoring** environment to use the *controller strategy*. Do not make this change unless a specific need for it exists; if you decide to enable controller strategy, make this change on Business Central and on all KIE Servers in the same environment.

NOTE

Do not enable the controller strategy in an environment with a high-availability Business Central. In such environments the controller strategy does not function correctly.

To enable the controller strategy on Business Central, set the **KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED** environment variable to **false**.

Next steps

If you want to deploy the environment with the default configuration of KIE Servers, click **Finish** and then click **Deploy** to deploy the environment. Otherwise, continue to set configuration parameters for KIE Servers.

3.2.5. Setting custom KIE Server configuration of the environment

Every environment type in the Business Automation operator includes one or several KIE Servers by default.

Optionally, you can set custom configuration for KIE Servers. In this case, default KIE Servers are not created and only the KIE Servers that you configure are deployed.

Prerequisites

- You completed basic configuration of a Red Hat Decision Manager environment using the Business Automation operator in the installer wizard according to the instructions in Section 3.2.2, “Setting the basic configuration of the environment” .
- If you want to use RH-SSO or LDAP for authentication, you completed security configuration according to the instructions in Section 3.2.3, “Setting the security configuration of the environment”.

Procedure

1. If the **Installation**, **Security**, or **Console** tab is open, click **Next** until you view the **KIE Servers** tab.

2. Click **Add new KIE Server** to add a new KIE Server configuration.

3. In the **Id** field, enter an identifier for the KIE Server. If the KIE Server connects to a Business Central or Business Central Monitoring instance, this identifier determines which server group the server joins.

4. In the **Name** field, enter a name for the KIE Server.

5. In the **Deployments** field, enter the number of similar KIE Servers that are to be deployed. The installer can deploy several KIE Servers with the same configuration. The identifiers and names of the KIE Servers are modified automatically and remain unique.

6. If you created the secret for KIE Server according to the instructions in Section 2.2, "Creating the secrets for KIE Server", enter the name of the secret in the **Keystore secret** field.

7. Optional: Enter the number of replicas for the KIE Server deployment in the **Replicas** field.

8. Optional: If you want to use a custom image for the KIE Server deployment, complete one the following sets of additional steps:

    a. If you want to use a Docker image by specifying the image in the registry:

        i. Set the custom registry in the **Installation** tab. If you do not set the custom registry, the installation uses the default Red Hat registry. For more information about setting the custom registry value, see Section 3.2.2, "Setting the basic configuration of the environment".

        ii. In the **KIE Server** tab, set the following fields:

        - **Image context**: The context of the image in the registry.

        - **Image**: The name of the image.

        - **Image tag**: The tag of the image. If you do not set this field, the installation uses the **latest** tag.
          For example, if the full address of the image is **registry.example.com/mycontext/myserver:1.0-SNAPSHOT**, set the custom registry to **registry.example.com**, the **Image context** field to **mycontext**, the **Image** field to **myserver**, and the **Image tag** field to **1.0-SNAPSHOT**.

    b. If you want to use an image from an existing OpenShift image stream:

        i. Click **Set KIE Server image**

        ii. Enter the name of the image stream tag in the **Name** field.

        iii. If the image stream is not in the **openshift** namespace, enter the namespace in the **Namespace** field.
        If the image stream tag is already configured in your OpenShift environment, the installation uses this tag. If the tag is not configured, the installation creates an image stream tag with the default image names and tags.

NOTE

Do not change the **Kind** value to **DockerImage**. This option does not work in Red Hat Decision Manager 7.9.1.

For instructions about creating custom images, see Section 3.5, "Creating custom images for KIE Server".

9. If you want to configure an immutable KIE Server using a Source to Image (S2I) build, complete the following additional steps:

IMPORTANT

If you want to configure an immutable KIE Server that pulls services from the Maven repository, do not click **Set Immutable server configuration** and do not complete these steps. Instead, set the **KIE_SERVER_CONTAINER_REPLOYMENT** environment variable.

a. Click **Set Immutable server configuration**.

b. In the **KIE Server container deployment** field, enter the identifying information of the services (KJAR files) that the deployment must extract from the result of a Source to Image (S2I) build. 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**.

c. If your OpenShift environment does not have a connection to the public Internet, enter the URL of the Maven mirror that you set up according to Section 2.9, "Preparing a Maven mirror repository for offline use" in the **Maven mirror URL** field.

d. In the **Artifact directory** field, enter 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.

e. If you want to use a custom base KIE Server image for the S2I build, click **Set Base build image** and then enter the name of the image stream in the **Name** field. If the image stream is not in the **openshift** namespace, enter the namespace in the **Namespace** field. If you want to use a Docker image name and not an OpenShift image stream tag, change the **Kind** value to **DockerImage**.

f. Click **Set Git source** and enter information in the following fields:

- **S2I Git URI**: The URI for the Git repository that contains the source for your services.
- **Reference**: The branch in the Git repository.
- **Context directory**: (Optional) The path to the source within the project downloaded from the Git repository. By default, the root directory of the downloaded project is the source directory.

> NOTE
>
> If you do not configure a Git source, the immutable KIE Server does not use an S2I build. Instead, it pulls the artifacts that you define in the **KIE Server container deployment** field from the configured Maven repository.

g. If you are using S2I and want to set a Git Webhook so that changes in the Git repository cause an automatic rebuild of the KIE Server, click **Add new Webhook**. Then select the type of the Webhook in the **Type** field and enter the secret string for the Webhook in the **Secret** field.

h. If you want to set a build environment variable for the S2I build, click **Add new Build Config Environment variable** and then enter the name and value for the variable in the **Name** and **Value** fields.

10. Optionally, enter requested and maximum CPU and memory limits in the fields under **Resource quotas**. If you are configuring several KIE Servers, the limits apply to each server separately.

11. If you selected RH-SSO authentication, configure RH-SSO for the KIE Server:

    a. Enter the client name in the **Client name** field and the client secret in the **Client secret** field. If a client with this name does not exist, the deployment attempts to create a new client with this name and secret.

    b. If the deployment is to create a new client, enter the HTTP and HTTPS URLs that will be used for accessing the KIE Server into the **SSO HTTP URL** and **SSO HTTPS URL** fields. This information is recorded in the client.

12. If you want to interact with the KIE Server through JMS API using an external AMQ message broker, enable the **Enable JMS Integration** setting. Additional fields for configuring JMS Integration are displayed and you must enter the values as necessary:

    - **User name**, **Password**: The user name and password of a standard broker user, if user authentication in the broker is required in your environment.

    - **Executor**: Select this setting to disable the JMS executor. The executor is enabled by default.

    - **Executor transacted**: Select this setting to enable JMS transactions on the executor queue.

    - **Enable signal**: Select this setting to enable signal configuration through JMS.

    - **Enable audit** Select this setting to enable audit logging through JMS.

    - **Audit transacted**: Select this setting to enable JMS transactions on the audit queue.

    - **Queue executor**, **Queue request**, **Queue response**, **Queue signal**, **Queue audit** Custom JNDI names of the queues to use. If you set any of these values, you must also set the **AMQ queues** parameter.

    - **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 are using any custom queue names, you must enter the names of all the queues uses by the server in this field.

- **Enable SSL integration**: Select this setting if you want to use an SSL connection to the AMQ broker. In this case you must also provide the name of the secret that you created in Section 2.4, "Creating the secrets for the AMQ broker connection" and the names and passwords of the key store and trust store that you used for the secret.

13. If you want to customize the configuration of the Java virtual machine on the KIE Server pods, select the **Enable JVM configuration** box and then enter information in any of the fields under **Enable JVM configuration**. All fields are optional. For the JVM parameters that you can configure, see Section 3.4, "JVM configuration parameters".

14. Optionally, depending on your needs, set environment variables. To set an environment variable, click **Add new Environment variable**, then enter the name and value for the variable in the **Name** and **Value** fields.

    - If you want to configure an immutable KIE server that pulls services from the configured Maven repository, enter the following settings:

        i. Set the **KIE_SERVER_CONTAINER_DEPLOYMENT** environment variable. The variable must contain the identifying information of the 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**.

        ii. Configure an external Maven repository.

    - If you want to configure an external Maven repository, set the following variables:

        - **MAVEN_REPO_URL**: The URL for the Maven repository

        - **MAVEN_REPO_ID**: An identifier for the Maven repository, for example, **repo-custom**

        - **MAVEN_REPO_USERNAME**: The user name for the Maven repository

        - **MAVEN_REPO_PASSWORD**: The password for the Maven repository

    - If your OpenShift environment does not have a connection to the public Internet, configure access to a Maven mirror that you set up according to Section 2.9, "Preparing a Maven mirror repository for offline use". Set the following variables:

        - **MAVEN_MIRROR_URL**: The URL for the Maven mirror repository that you set up in Section 2.9, "Preparing a Maven mirror repository for offline use". This URL must be accessible from a pod in your OpenShift environment. If you configured this KIE Server as S2I, you already entered this URL.

        - **MAVEN_MIRROR_OF**: The value that determines which artifacts are to be retrieved from the mirror. If you configured this KIE Server as S2I, do not set this value. 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 your authoring environment uses a built-in Business Central Maven repository, If your authoring environment uses a built-in Business Central Maven repository, change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror: **external:*,!repo-rhdmcentr**.

- If you want to configure your KIE Server deployment to use Prometheus to collect and store metrics, set the **PROMETHEUS_SERVER_EXT_DISABLED** environment variable to **false**. For instructions about configuring Prometheus metrics collection, see *Managing and monitoring KIE Server*.

- If you are using Red Hat Single Sign-On authentication and the interaction of your application with Red Hat Single Sign-On requires support for CORS, set the **SSO_ENABLE_CORS** variable to **true**.

- In some authoring environments, you might need to ensure that several users can deploy services on the same KIE Server at the same time. By default, after deploying a service onto a KIE Server using Business Central, the user needs to wait for some seconds before more services can be deployed. The **OpenShiftStartupStrategy** setting is enabled by default and causes this limitation. To remove the limitation, you can configure an **rhdm-authoring** environment to use the *controller strategy*. Do not make this change unless a specific need for it exists; if you decide to enable controller strategy, make this change on Business Central and on all KIE Servers in the same environment.

  > NOTE
  >
  > Do not enable the controller strategy in an environment with a high-availability Business Central. In such environments the controller strategy does not function correctly.

  To enable controller strategy on a KIE Server, set the **KIE_SERVER_STARTUP_STRATEGY** environment variable to **ControllerBasedStartupStrategy** and the **KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED** environment variable to **false**.

Next steps

To configure additional KIE Servers, click **Add new KIE Server** again and repeat the procedure for the new server configuration.

Click **Finish** and then click **Deploy** to deploy the environment.

3.3. MODIFYING AN ENVIRONMENT THAT IS DEPLOYED USING OPERATORS

If an environment is deployed using operators, you cannot modify it using typical OpenShift methods. For example, if you delete a deployment configuration or a service, it is re-created automatically with the same parameters.

To modify the environment, you must modify the YAML description of the environment. You can change common settings such as passwords, add new KIE Servers, and scale KIE Servers.

Procedure

1. Enter your project in the OpenShift web cluster console.

2. In the OpenShift Web console navigation panel, select **Catalog → Installed operators** or **Operators → Installed operators**.

3. Find the **Business Automation** operator line in the table and click **KieApp** in the line. Information about the environments that you deployed using this operator is displayed.

4. Click the name of a deployed environment.

5. Select the **YAML** tab.
   A YAML source is displayed. In this YAML source, you can edit the content under **spec:** to change the configuration of the environment.

6. If you want to change the deployed version of Red Hat Decision Manager, add the following line under **spec:**

   ```
   version: 7.9.1
   ```

   You can replace **7.9.1** with another required version. Use this setting to upgrade Red Hat Decision Manager to a new version if automatic updates are disabled, for example, if you use a custom image.

7. If you want to change common settings, such as passwords, edit the values under **commonConfig:**.

8. If you want to add new KIE Servers, add their descriptions at the end of the block under **servers:**, as shown in the following examples:

   - To add two servers named **server-a** and **server-a-2**, add the following lines:

     ```
     - deployments: 2
       name: server-a
     ```

   - To add an immutable KIE Server that includes services built from source in an S2I process, add the following lines:

     ```
     - build:
         kieServerContainerDeployment: <deployment>
         gitSource:
           uri: <url>
           reference: <branch>
           contextDir: <directory>
     ```

     Replace the following values:

     - **<deployment>**: The identifying information of the decision service (KJAR file) that is built from your source. The format is **<containerId>=<groupId>:<artifactId>:<version>**. You can provide two or more KJAR files using the **|** separator, for example **containerId=groupId:artifactId:version|c2=g2:a2:v2**. The Maven build process must produce all these files from the source in the Git repository.

     - **<url>**: The URL for the Git repository that contains the source for your decision service.

     - **<branch>**: The branch in the Git repository.

     - **<directory>**: The path to the source within the project downloaded from the Git repository.

9. If you want to scale a KIE Server, find the description of the server in the block under **servers:** and add a **replicas:** setting under that description. For example, **replicas: 3** scales the server to three pods.

10. If you want to make other changes, review the CRD source for the available settings. To view the CRD source, log in to the Red Hat OpenShift Container Platform environment with the **oc** command as an administrative user and then enter the following command:

```
oc get crd kieapps.app.kiegroup.org -o yaml
```

11. Click **Save** and then wait for a **has been updated** pop-up message.

12. Click **Reload** to view the new YAML description of the environment.

3.4. JVM CONFIGURATION PARAMETERS

When deploying Red Hat Decision Manager using the operator, you can optionally set a number of JVM configuration parameters for Business Central and KIE Servers. These parameters set environment variables for the corresponding containers.

The following table lists all JVM configuration parameters that you can set when deploying Red Hat Decision Manager using the operator.

The default settings are optimal for most use cases. Make any changes only when they are required.

Table 3.1. JVM configuration parameters

Configuration field	Environment variable	Description	Example
Java Opts append	JAVA_OPTS_APPEND	User specified Java options to be appended to generated options in JAVA_OPTS.	**-Dsome.property=foo**
Java max memory ratio	JAVA_MAX_MEM_RATIO	The maximum percentage of container memory that can be used for the Java Virtual Machine. The remaining memory is used for the operating system. The default value is **50**, for a limit of 50%. Sets the **-Xmx** JVM option. If you enter a value of **0**, the **-Xmx** option is not set.	**40**
Java initial memory ratio	JAVA_INITIAL_MEM_RATIO	The percentage of container memory that is initially used for the Java Virtual Machine. The default value is **25**, so 25% of the pod memory is initially allocated for the JVM if this value does not exceed the **Java Max Initial Memory** value. Sets the **-Xms** JVM option. If you enter a value of **0**, the **-Xms** option is not set.	**25**

Configuration field	Environment variable	Description	Example
Java max initial memory	JAVA_MAX_INITIAL_MEM	The maximum amount of memory, in megabytes, that can be initially used for the Java Virtual Machine. If the initial allocated memory, as set in the **Java initial memory ratio** parameter, would otherwise be greater than this value, the amount of memory set in this value is allocated using the **-Xms** JVM option. The default value is **4096**.	**4096**
Java diagnostics	JAVA_DIAGNOSTICS	Enable this setting to enable output of additional JVM diagnostic information to the standard output. Disabled by default.	**true**
Java debug	JAVA_DEBUG	Enable this setting to switch on remote debugging. Disabled by default. Adds the **-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=${debug_port}** JVM option, where **${debug_port}** defaults to **5005**.	**true**
Java debug port	JAVA_DEBUG_PORT	The port that is used for remote debugging. The default value is **5005**.	**8787**
GC min heap free ratio	GC_MIN_HEAP_FREE_RATIO	Minimum percentage of heap free after garbage collection (GC) to avoid expansion. Sets the **-XX:MinHeapFreeRatio** JVM option.	**20**
GC max heap free ratio	GC_MAX_HEAP_FREE_RATIO	Maximum percentage of heap free after GC to avoid shrinking. Sets the **-XX:MaxHeapFreeRatio** JVM option.	**40**
GC time ratio	GC_TIME_RATIO	Specifies the ratio of the time spent outside the garbage collection (for example, the time spent for application execution) to the time spent in the garbage collection. Sets the **-XX:GCTimeRatio** JVM option.	**4**
GC adaptive size policy weight	GC_ADAPTIVE_SIZE_POLICY_WEIGHT	The weighting given to the current GC time versus previous GC times. Sets the **-XX:AdaptiveSizePolicyWeight** JVM option.	**90**
GC max metaspace size	GC_MAX_METASPACE_SIZE	The maximum metaspace size. Sets the **-XX:MaxMetaspaceSize** JVM option.	**100**

3.5. CREATING CUSTOM IMAGES FOR KIE SERVER

You can create custom images to add files to KIE Server deployments. You must push the images to your own container registry. When deploying Red Hat Decision Manager, you can configure the operator to use the custom images.

If you use a custom image, you must disable automatic version updates. When you want to install a new version, build the image with the same name as before and the new version tag and push the image into your registry. You can then change the version and the operator automatically pulls the new image. For instructions about changing the product version in the operator, see Section 3.3, “Modifying an environment that is deployed using operators”.

In particular, you can create the following types of custom images:

- A custom image of KIE Server that includes an additional RPM package
- A custom image of KIE Server that includes an additional JAR class library

3.5.1. Creating a custom KIE Server image with an additional RPM package

You can create a custom KIE Server image where an additional RPM package is installed. You can push this image into your custom registry and then use it to deploy the KIE Server.

You can install any package from the Red Hat Enterprise Linux 8 repository. This example installs the **procps-ng** package, which provides the **ps** utility, but you can modify it to install other packages.

Procedure

1. Authenticate to the **registry.redhat.io** registry using the **podman login** command. For instructions about authenticating to the registry, see Red Hat Container Registry Authentication.

2. To download the supported KIE Server base image, enter the following command:

   ```
   podman pull registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.9.1
   ```

3. Create a **Dockerfile** that defines a custom image based on the base image. The file must change the current user to **root**, install the RPM package using the **yum** command, and then revert to **USER 185**, the Red Hat JBoss EAP user. The following example shows the content of the **Dockerfile** file:

   ```
   FROM registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.9.1
   USER root
   RUN yum -y install procps-ng
   USER 185
   ```

   Replace the name of the RPM file as necessary. The **yum** command automatically installs all dependencies from the Red Hat Enterprise Linux 8 repository. You might need to install several RPM files, in this case, use several **RUN** commands.

4. Build the custom image using the **Dockerfile**. Supply the fully qualified name for the image, including the registry name. You must use the same version tag as the version of the base image. To build the image, enter the following command:

   ```
   podman build . --tag registry_address/image_name:7.9.1
   ```

For example:

```
podman build . --tag registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

5. After the build completes, run the image, log in to it, and verify that the customization was successful. Enter the following command:

```
podman run -it --rm registry_address/image_name:7.9.1 /bin/bash
```

For example:

```
podman run -it --rm registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1 /bin/bash
```

In the shell prompt for the image, enter the command to test that the RPM is installed, then enter **exit**. For example, for **procps-ng**, run the **ps** command:

```
[jboss@c2fab36b778e ~]$ ps
PID TTY          TIME CMD
  1 pts/0    00:00:00 bash
 13 pts/0    00:00:00 ps
[jboss@c2fab36b778e ~]$ exit
```

6. To push the custom image into your registry, enter the following command:

```
podman push registry_address/image_name:7.9.1
docker://registry_address/image_name:7.9.1
```

For example:

```
podman push registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
docker://registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
```

Next steps

When deploying the KIE Server, set the image name and namespace to specify the custom image in your registry. Click **Set KIE Server image**, change the **Kind** value to **DockerImage**, and then provide the image name including the registry name, but without the version tag, for example:

```
registry.example.com/custom/rhdm-kieserver-rhel8
```

For instructions about deploying the KIE Server using the operator, see Section 3.2.5, "Setting custom KIE Server configuration of the environment".

3.5.2. Creating a custom KIE Server image with an additional JAR file

You can create a custom KIE Server image where an additional JAR file (or several JAR files) is installed to extend the capabilities of the server. You can push this image into your custom registry and then use it to deploy the KIE Server.

For example, you can create a custom class JAR to provide custom Prometheus metrics in the KIE Server. For instructions about creating the custom class, see Extending Prometheus metrics monitoring in KIE Server with custom metrics in *Managing and monitoring KIE Server*.

Procedure

1. Develop a custom library that works with the KIE Server. You can use the following documentation and examples to develop the library:
   - KIE Server capabilities and extensions in *Managing and monitoring KIE Server*.
   - Domain-specific Prometheus metrics with Red Hat Process Automation Manager and Decision Manager
   - Extend KIE Server with additional transport
2. Build the library using Maven, so that the JAR file is placed in the **target** directory. This example uses the **custom-kieserver-ext-1.0.0.Final.jar** file name.
3. Authenticate to the **registry.redhat.io** registry using the **podman login** command. For instructions about authenticating to the registry, see Red Hat Container Registry Authentication.
4. To download the supported KIE Server base image, enter the following command:

   ```
   podman pull registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.9.1
   ```

5. Create a **Dockerfile** that defines a custom image based on the base image. The file must copy the JAR file (or several JAR files) into the **/opt/eap/standalone/deployments/ROOT.war/WEB-INF/lib/** directory. The following example shows the content of the **Dockerfile** file:

   ```
   FROM registry.redhat.io/rhdm-7/rhdm-kieserver-rhel8:7.9.1
   COPY target/custom-kieserver-ext-1.0.0.Final.jar
   /opt/eap/standalone/deployments/ROOT.war/WEB-INF/lib/
   ```

6. Build the custom image using the **Dockerfile**. Supply the fully qualified name for the image, including the registry name. You must use the same version tag as the version of the base image. To build the image, enter the following command:

   ```
   podman build . --tag registry_address/image_name:7.9.1
   ```

   For example:

   ```
   podman build . --tag registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
   ```

7. To push the custom image into your registry, enter the following command:

   ```
   podman push registry_address/image_name:7.9.1
   docker://registry_address/image_name:7.9.1
   ```

   For example:

   ```
   podman push registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
   docker://registry.example.com/custom/rhdm-kieserver-rhel8:7.9.1
   ```

Next steps

When deploying the KIE Server, set the image name and namespace to specify the custom image in your registry. Click **Set KIE Server image**, change the **Kind** value to **DockerImage**, and then provide the image name including the registry name, but without the version tag, for example:

```
registry.example.com/custom/rhdm-kieserver-rhel8
```

For instructions about deploying the KIE Server using the operator, see Section 3.2.5, “Setting custom KIE Server configuration of the environment”.

CHAPTER 4. MIGRATION OF INFORMATION FROM A DEPLOYMENT ON RED HAT OPENSHIFT CONTAINER PLATFORM VERSION 3

If you previously used a Red Hat Decision Manager deployment on Red Hat OpenShift Container Platform version 3, you can migrate the information from that deployment to a new deployment on Red Hat OpenShift Container Platform version 4.

Before migrating information, you must deploy a new Red Hat Decision Manager infrastructure on Red Hat OpenShift Container Platform version 4 using the operator. Include the same elements in the new infrastructure as those present in the old deployment. For example:

- For any existing authoring deployment, create a new authoring infrastructure, including Business Central and at least one KIE Server.
- For any existing immutable KIE Server, deploy a new immutable KIE Server with the same artifacts.

4.1. MIGRATING INFORMATION IN BUSINESS CENTRAL

If you have an existing authoring environment in Red Hat OpenShift Container Platform version 3, you can copy the **.niogit** repository and the Maven repository from Business Central in this environment to Business Central in a new deployment on Red Hat OpenShift Container Platform version 4. This action makes all the same projects and artifacts available in the new deployment.

Prerequisites

- You must have a machine that has network access to both the Red Hat OpenShift Container Platform version 3 and Red Hat OpenShift Container Platform version 4 infrastructures.
- The **oc** command-line client from Red Hat OpenShift Container Platform version 4 must be installed on the machine. For instructions about installing the command-line client, see *CLI tools* in Red Hat OpenShift Container Platform documentation.

Procedure

1. Ensure that no web clients and no client applications are connected to any elements of the old and new deployment, including Business Central and KIE Servers.
2. Create an empty temporary directory and change into it.
3. Using the **oc** command, log in to the Red Hat OpenShift Container Platform version 3 infrastructure and switch to the project containing the old deployment.
4. To view the pod names in the old deployment, run the following command:

   ```
   oc get pods
   ```

   Find the Business Central pod. The name of this pod includes **rhdmcentr**. In a high-availability deployment, you can use any of the Business Central pods.
5. Use the **oc** command to copy the the **.niogit** repository and the Maven repository from the pod to the local machine, for example:

```
oc cp myapp-rhdmcentr-5-689mw:/opt/kie/data/.niogit .niogit
oc cp myapp-rhdmcentr-5-689mw:/opt/kie/data/maven-repository maven-repository
```

6. Using the **oc** command, log in to the Red Hat OpenShift Container Platform version 4 infrastructure and switch to the project containing the new deployment.

7. To view the pod names in the new deployment, run the following command:

   ```
   oc get pods
   ```

   Find the Business Central pod. The name of this pod includes **rhdmcentr**. In a high-availability deployment, you can use any of the Business Central pods.

8. Use the **oc** command to copy the the **.niogit** repository and the Maven repository from the local machine to the pod, for example:

   ```
   oc cp .niogit myappnew-rhdmcentr-abd24:/opt/kie/data/.niogit
   oc cp maven-repository myappnew-rhdmcentr-abd24:/opt/kie/data/maven-repository
   ```

PART II. DEPLOYING A RED HAT DECISION MANAGER ENVIRONMENT ON RED HAT OPENSHIFT CONTAINER PLATFORM USING TEMPLATES

As a system engineer, you can deploy a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform version 4 to provide an infrastructure to develop or execute services and other business assets. You can use one of the supplied templates to deploy a predefined Red Hat Decision Manager environment to suit your particular needs.

Prerequisites

- Red Hat OpenShift Container Platform version 3.11 is deployed.
- The following resources are available on the OpenShift cluster. Depending on the application load, higher resource allocation might be necessary for acceptable performance.
  - For an authoring environment, 4 gigabytes of memory and 2 virtual CPU cores for the Business Central pod. In a high-availability deployment, these resources are required for each replica and two replicas are created by default.
  - 2 gigabytes of memory and 1 virtual CPU core for each replica of each KIE Server pod.
  - In a high-availability authoring deployment, additional resources according to the configured defaults are required for the Red Hat AMQ, and Red Hat Data Grid pods.
- Dynamic persistent volume (PV) provisioning is enabled. Alternatively, if dynamic PV provisioning is not enabled, enough persistent volumes must be available. By default, the deployed components require the following PV sizes:
  - By default, Business Central requires one 1Gi PV. You can change the PV size for Business Central persistent storage.

> **NOTE**
>
> For instructions about checking the capacity of your cluster, see Analyzing cluster capacity in the Red Hat OpenShift Container Platform 3.11 product documentation.

- The OpenShift project for the deployment is created.
- You are logged into 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.
- Dynamic persistent volume (PV) provisioning is enabled. Alternatively, if dynamic PV provisioning is not enabled, enough persistent volumes must be available. By default, the deployed components require the following PV sizes:
  - The replicated set of KIE Server pods requires one 1Gi PV for the database by default. You can change the database PV size in the template parameters. This requirement does not apply if you use an external database server.
  - Business Central requires one 1Gi PV by default. You can change the PV size for Business Central persistent storage in the template parameters.
- If you intend to scale any of the Business Central pods, your OpenShift environment supports

persistent volumes with **ReadWriteMany** mode. If your environment does not support this mode, you can use NFS to provision the volumes. However, for best performance and reliability, use GlusterFS to provision persistent volumes for a high-availability authoring environment. For information about access mode support in OpenShift public and dedicated clouds, see Access Modes.

NOTE

Since Red Hat Decision Manager version 7.5, images and templates for Red Hat OpenShift Container Platform 3.x are deprecated. These images and templates do not get new features, but remain supported until the end of full support for Red Hat OpenShift Container Platform version 3.x. For more information about the full support lifecycle phase for Red Hat OpenShift Container Platform version 3.x, see Red Hat OpenShift Container Platform Life Cycle Policy (non-current versions).

NOTE

Do not use Red Hat Decision Manager templates with Red Hat OpenShift Container Platform 4.x. To deploy Red Hat Decision Manager on Red Hat OpenShift Container Platform 4.x, see the instructions in *Deploying a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform using Operators*.

CHAPTER 5. 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:

- KIE Server, also known as *Execution 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.
  In some templates, you can scale up a KIE 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 KIE 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 KIE 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 KIE 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.

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

The following environment types are typical:

- *Trial*: an environment for demonstration and evaluation of Red Hat Decision Manager. This environment includes Business Central and a KIE 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.

- *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 KIE Servers. It consists of pods that provide Business Central for the authoring work and one or more KIE Servers for execution of the services. Each KIE Server is a pod that you can replicate by scaling it up or down as necessary. You can deploy and undeploy services on each KIE Server using Business Central.

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

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

5.1. ARCHITECTURE OF AN AUTHORING ENVIRONMENT

In Red Hat Decision Manager, the Business Central component provides a web-based interactive user interface for authoring services. The KIE Server component runs the services.

You can also use Business Central to deploy services onto a KIE Server. You can use several KIE Servers to run different services and control the servers from the same Business Central.

Single authoring environment

In a single authoring environment, only one instance of Business Central is running. Multiple users can access its web interface at the same time, however the performance can be limited and there is no failover capability.

Business Central includes a built-in Maven repository that stores the built versions of the services that you develop (KJAR files/artifacts). You can use your continuous integration and continuous deployment (CICD) tools to retrieve these artifacts from the repository and move them as necessary.

Business Central saves the source code in a built-in Git repository, stored in the **.niogit** directory. It uses a built-in indexing mechanism to index the assets in your services.

Business Central uses persistent storage for the Maven repository and for the Git repository.

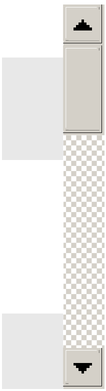
A single authoring environment, by default, includes one KIE Server.

A single authoring environment, by default, uses the *controller strategy*. Business Central includes the *Controller*, a component that can manage KIE Servers. When you configure a KIE Server to connect to Business Central, the KIE Server uses a REST API to connect to the Controller. This connection opens a persistent WebSocket. In an OpenShift deployment that uses the controller strategy, each KIE Server is initially configured to connect to the Business Central Controller.

When you use the Business Central user interface to deploy or manage a service on the KIE Server, the KIE Server receives the request through the Controller connection WebSocket. To deploy a service, the KIE Server requests the necessary artifact from the Maven repository that is a part of Business Central.

Client applications use a REST API to use services that run on the KIE Server.

Figure 5.1. Architecture diagram for a single authoring environment



Clustering KIE Servers and using multiple KIE Servers

You can scale a KIE Server pod to run a clustered KIE Server environment.

In a clustered deployment, several instances of the KIE Server run the same services. These servers can connect to the Business Central Controller using the same server ID, so they can receive the same requests from the controller. Red Hat OpenShift Container Platform provides load-balancing between the servers. The services that run on a clustered KIE Server must be stateless, because requests from the same client might be processed by different instances.

You can also deploy several independent KIE Servers to run different services. In this case, the servers connect to the Business Central Controller with different server ID values. You can use the Business Central UI to deploy services to each of the servers.

Smart Router

The optional Smart Router component provides a layer between client applications and KIE Servers. It can be useful if you are using several independent KIE Servers.

The client application can use services running on different KIE Servers, but always connects to the Smart Router. The Smart Router automatically passes the request to the KIE Servers that runs the required service. The Smart Router also enables management of service versions and provides an additional load-balancing layer.

High-availability authoring environment

In a high-availability (HA) authoring environment, the Business Central pod is scaled, so several instances of Business Central are running. Red Hat OpenShift Container Platform provides load balancing for user requests. This environment provides optimal performance for multiple users and supports failover.

Each instance of Business Central includes the Maven repository for the built artifacts and uses the **.niogit** Git repository for source code. The instances use shared persistent storage for the repositories. A persistent volume with **ReadWriteMany** access is required for this storage.

An instance of Red Hat DataGrid provides indexing of all projects and assets developed in Business Central.

An instance of Red Hat AMQ propagates Java CDI messages between all instances of Business Central. For example, when a new project is created or when an asset is locked or modified on one of the instances, this information is immediately reflected in all other instances.

The controller strategy is not suitable for clustered deployment. In an OpenShift deployment, a high-availability Business Central must manage KIE Servers using the *OpenShift startup strategy*.

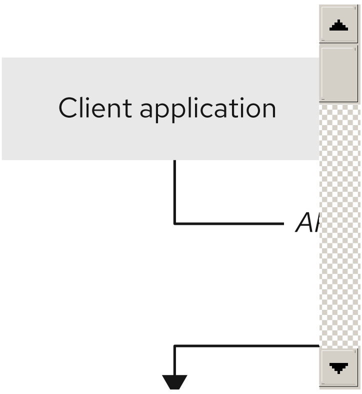
Each KIE Server deployment (which can be scaled) creates a ConfigMap that reflects its current state. The Business Central discovers all KIE Servers by reading their ConfigMaps.

When the user requests a change in KIE Server configuration (for example, deploys or undeploys a service), Business Central initiates a connection to the KIE Server and sends a REST API request. The KIE Server changes the ConfigMap to reflect the new configuration state and then triggers its own redeployment, so that all instances are redeployed and reflect the new configuration.

You can deploy several independent KIE Servers in your OpenShift environment. Each of the KIE Servers has a separate ConfigMap with the necessary configuration. You can scale each of the KIE Servers separately.

You can include Smart Router in the OpenShift deployment.

Figure 5.2. Architecture diagram for a high-availability authoring environment



CHAPTER 6. PREPARATION FOR DEPLOYING RED HAT DECISION MANAGER IN YOUR OPENSHIFT ENVIRONMENT

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

NOTE

If you are deploying a trial environment, complete the procedure described in Section 6.1, “Ensuring the availability of image streams and the image registry” and do not complete any other preparation procedures.

6.1. 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 -F rhdm79-decisioncentral-openshift
   $ oc get imagestreamtag -n openshift | grep -F rhdm79-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-7.9.1-openshift-templates.zip** product deliverable file from the Software Downloads page and extract the **rhdm79-image-streams.yaml** file.

g. Enter the following command:

```
$ oc apply -f rhdm79-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.

6.2. CREATING THE SECRETS FOR KIE 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 Red Hat OpenShift Container Platform documentation.

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

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 KIE Server.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. 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
   ```

6.3. CREATING THE SECRETS FOR BUSINESS CENTRAL

If your environment includes Business Central, you must create an SSL certificate for HTTP access to Business Central and provide it to your OpenShift environment as a secret.

Do not use the same certificate and keystore for Business Central and KIE Server.

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 Business Central.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. Use the **oc** command to generate a secret named **decisioncentral-app-secret** from the new keystore file:

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

6.4. CREATING THE SECRETS FOR SMART ROUTER

If your environment includes Smart Router, you must create an SSL certificate for HTTP access to Smart Router and provide it to your OpenShift environment as a secret.

Do not use the same certificate and keystore for Smart Router as the ones used for KIE Server or Business Central.

Procedure

1. Generate an SSL keystore named **keystore.jks** with a private and public key for SSL encryption for KIE 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 Smart Router.

2. Record the name of the certificate. The default value for this name in Red Hat Decision Manager configuration is **jboss**.

3. Record the password of the keystore file. The default value for this name in Red Hat Decision Manager configuration is **mykeystorepass**.

4. Use the **oc** command to generate a secret named **smartrouter-app-secret** from the new keystore file:

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

6.5. CREATING THE SECRET FOR THE ADMINISTRATIVE USER

You must create a generic secret that contains the user name and password for a Red Hat Decision Manager administrative user account. This secret is required for deploying Red Hat Decision Manager using any template except the trial template.

The secret must contain the user name and password as literals. The key name for the user name is **KIE_ADMIN_USER**. The key name for the password is **KIE_ADMIN_PWD**.

If you are using multiple templates to deploy components of Red Hat Decision Manager, use the same secret for all these deployments. The components utilize this user account to communicate with each other.

If your environment includes Business Central, you can also use this user account to log in to Business Central.

IMPORTANT

If you use RH-SSO or LDAP authentication, the same user with the same password must be configured in your authentication system with the **kie-server,rest-all,admin** roles for Red Hat Decision Manager.

Procedure

Use the **oc** command to generate a generic secret named **kie-admin-user-secret** from the user name and password:

```
$ oc create secret generic rhpam-credentials --from-literal=KIE_ADMIN_USER=adminUser --from-
literal=KIE_ADMIN_PWD=adminPassword
```

In this command, replace *adminPassword* with the password for the administrative user. Optionally, you can replace *adminUser* with another user name for the administrative user.

6.6. CHANGING GLUSTERFS CONFIGURATION

If you are deploying an authoring environment, you must check whether your OpenShift environment uses GlusterFS to provide permanent storage volumes. If it uses GlusterFS, to ensure optimal performance of Business Central, you must tune your GlusterFS storage by changing the storage class configuration.

Procedure

1. To check whether your environment uses GlusterFS, enter the following command:

```
oc get storageclass
```

In the results, check whether the **(default)** marker is on the storage class that lists **glusterfs**. For example, in the following output the default storage class is **gluster-container**, which does list **glusterfs**:

```
NAME              PROVISIONER               AGE
gluster-block     gluster.org/glusterblock  8d
gluster-container (default) kubernetes.io/glusterfs 8d
```

If the result has a default storage class that does not list **glusterfs** or if the result is empty, you do not need to make any changes. In this case, skip the rest of this procedure.

2. To save the configuration of the default storage class into a YAML file, enter the following command:

   ```
   oc get storageclass <class-name> -o yaml >storage_config.yaml
   ```

   Replace **<class-name>** with the name of the default storage class. Example:

   ```
   oc get storageclass gluster-container -o yaml >storage_config.yaml
   ```

3. Edit the **storage_config.yaml** file:

   a. Remove the lines with the following keys:

   - **creationTimestamp**
   - **resourceVersion**
   - **selfLink**
   - **uid**

   b. If you are planning to use Business Central only as a single pod, without high-availability configuration, on the line with the **volumeoptions** key, add the following options:

   ```
   features.cache-invalidation on
   performance.nl-cache on
   ```

   For example:

   volumeoptions: client.ssl off, server.ssl off, features.cache-invalidation on, performance.nl-cache on

   c. If you are planning to use Business Central in a high-availability configuration, on the line with the **volumeoptions** key, add the following options:

   ```
   features.cache-invalidation on
   nfs.trusted-write on
   nfs.trusted-sync on
   performance.nl-cache on
   performance.stat-prefetch off
   performance.read-ahead off
   performance.write-behind off
   ```

```
performance.readdir-ahead off
performance.io-cache off
performance.quick-read off
performance.open-behind off
locks.mandatory-locking off
performance.strict-o-direct on
```

For example:

volumeoptions: client.ssl off, server.ssl off, features.cache-invalidation on, nfs.trusted-write on, nfs.trusted-sync on, performance.nl-cache on, performance.stat-prefetch off, performance.read-ahead off, performance.write-behind off, performance.readdir-ahead off, performance.io-cache off, performance.quick-read off, performance.open-behind off, locks.mandatory-locking off, performance.strict-o-direct on

4. To remove the existing default storage class, enter the following command:

   ```
   oc delete storageclass <class-name>
   ```

   Replace **<class-name>** with the name of the default storage class. Example:

   ```
   oc delete storageclass gluster-container
   ```

5. To re-create the storage class using the new configuration, enter the following command:

   ```
   oc create -f storage_config.yaml
   ```

6.7. PROVISIONING PERSISTENT VOLUMES WITH READWRITEMANY ACCESS MODE USING NFS

If you want to deploy high-availability Business Central, your environment must provision persistent volumes with **ReadWriteMany** access mode. If you want to deploy high-availability Business Central, your environment must provision persistent volumes with **ReadWriteMany** access mode.

NOTE

If you want to deploy a high-availability authoring environment, for optimal performance and reliability, provision persistent volumes using GlusterFS. Configure the GlusterFS storage class as described in Section 6.6, "Changing GlusterFS configuration".

If your configuration requires provisioning persistent volumes with **ReadWriteMany** access mode but your environment does not support such provisioning, use NFS to provision the volumes. Otherwise, skip this procedure.

Procedure

Deploy an NFS server and provision the persistent volumes using NFS. For information about provisioning persistent volumes using NFS, see the "Persistent storage using NFS" section of the *Configuring Clusters* guide in the Red Hat OpenShift Container Platform 3.11 documentation.

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

If you are planning to create immutable KIE servers using the source-to-image (S2I) process, you must provide the source code for your services in a Git repository. 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.

Skip this procedure if you are not planning to use the S2I process or if you are not using Business Central for authoring services.

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.

6.9. 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. Configure a Maven release repository to which you have write access. The repository must allow read access without authentication and your OpenShift environment must have network 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 in the Red Hat OpenShift Container Platform 3.11 documentation.

   Use this repository as a mirror to host the publicly available Maven artifacts. You can also provide your own services in this repository in order to deploy these services on immutable servers.

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

   a. Click **Red Hat Process Automation Manager 7.9.1 Offliner Content List**to download the **rhdm-7.9.1-offliner.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

   b. Extract the contents of the **rhdm-7.9.1-offliner.zip** file into any directory.

   c. Change to the directory and enter the following command:

   ```
   ./offline-repo-builder.sh offliner.txt
   ```

   This command creates a **repository** subdirectory and downloads the necessary artifacts into this subdirectory.

   If a message reports that some downloads have failed, run the same command again. If downloads fail again, contact Red Hat support.

   d. Upload all artifacts from the **repository** subdirectory to the Maven mirror repository that you prepared. You can use the Maven Repository Provisioner utility, available from the Maven repository tools Git repository, 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, available from the Maven repository tools Git repository, to upload the artifacts.

CHAPTER 7. TRIAL ENVIRONMENT

You can deploy a trial (evaluation) Red Hat Decision Manager environment. It consists of Business Central for authoring or managing services and KIE Server for test execution of services.

This environment does not include permanent storage. Assets that you create or modify in a trial environment are not saved.

This environment is intended for test and demonstration access. It supports cross-origin resource sharing (CORS). This means that KIE Server endpoints can be accessed using a browser when other resources on the page are provided by other servers. KIE Server endpoints are normally intended for REST calls, but browser access can be needed in some demonstration configurations.

7.1. DEPLOYING A TRIAL ENVIRONMENT

The procedure to deploy a trial environment is minimal. There are no required settings and all passwords are set to a single value. The default password is **RedHat**.

Procedure

1. Download the **rhdm-7.9.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

2. Extract the **rhdm79-trial-ephemeral.yaml** template file.

3. Use one of the following methods to deploy the template:

   - In the OpenShift Web UI, select **Add to Project → Import YAML / JSON**and then select or paste the **rhdm79-trial-ephemeral.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>/rhdm79-trial-ephemeral.yaml
     ```

     In this command line, replace **<template-path>** with the path to the downloaded template file.

4. Optionally, set any parameters as described in the template. A typical trial deployment requires only the following parameter:

   - **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 6.1, “Ensuring the availability of image streams and the image registry”), the namespace is **openshift**. If you installed the image streams file, the namespace is the name of the OpenShift project.

5. Complete the creation of the environment, depending on the method that you are using:

   - In the OpenShift Web UI, click **Create**.

     - A **This will create resources that may have security or project behavior implications** pop-up message might be displayed. If it is displayed, click **Create Anyway**.

   - Complete and run the command line.

CHAPTER 8. AUTHORING OR MANAGED SERVER ENVIRONMENT

You can deploy an environment for creating and modifying services using Business Central and for running them in KIE Servers managed by Business Central. This environment consists of Business Central and one or more KIE Servers.

You can use Business Central both to develop services and to deploy them to KIE Servers. You can connect several KIE Servers to one Business Central to manage deployment of services to each of the servers.

If necessary, you can create separate environments, so that you can use one deployment of Business Central to author services (*authoring environment*) and another deployment of Business Central to manage deployment of staging or production services on several KIE Servers (*managed server environment*). Usually, one KIE Server is sufficient for a dedicated authoring environment. You can use an external Maven repository to store services from an authoring environment and deploy them to a separate managed server environment.

For Red Hat Decision Manager, the procedures to deploy an authoring environment and a managed server environment are the same. You must first deploy an authoring environment template, consisting of Business Central and one KIE Server.

If necessary, you can deploy additional KIE Server templates in the same namespace to create an environment with multiple KIE Servers. This environment can be a managed server environment for staging and production deployment of services.

Depending on your needs, you can deploy either a single authoring environment template or a high-availability (HA) authoring environment template.

A single authoring environment contains two pods. One of the pods runs Business Central, the other runs KIE Server. This environment is most suitable for single-user authoring or when your OpenShift infrastructure has limited resources. It does not require persistent volumes that support the **ReadWriteMany** access mode.

In a single authoring environment, you cannot scale Business Central. You can scale KIE Server.

In an HA authoring environment, both Business Central and KIE Server are provided in scalable pods. When pods are scaled, persistent storage is shared between the copies.

To enable high-availability functionality in Business Central, additional pods with AMQ and Data Grid are required. These pods are configured and deployed by the high-availability authoring template. Use a high-availability authoring environment to provide maximum reliability and responsiveness, especially if several users are involved in authoring at the same time.

In the current version of Red Hat Decision Manager, an HA authoring environment is supported with certain limitations:

- If a Business Central pod crashes while a user works with it, the user can get an error message and then is redirected to another pod. Logging on again is not required.
- If a Business Central pod crashes during a user operation, data that was not committed (saved) might be lost.
- If a Business Central pod crashes during creation of a project, an unusable project might be created.

- If a Business Central pod crashes during creation of an asset, the asset might be created but not indexed, so it cannot be used. The user can open the asset in Business Central and save it again to make it indexed.

- When a user deploys a service to the KIE Server, the KIE Server deployment is rolled out again. Users can not deploy another service to the same KIE Server until the roll-out completes.

In a high-availability authoring environment you can also deploy additional managed or immutable KIE Servers, if required. Business Central can automatically discover any KIE Servers in the same namespace, including immutable KIE Servers and managed KIE Servers.

If you want to deploy additional managed or immutable KIE Servers in a single authoring environment, you must complete an additional manual step to enable the **OpenShiftStartupStrategy** setting in the environment, as described in]. This setting enables the discovery of other KIE Servers.

For instructions about deploying managed KIE Servers, see Section 8.3, "Deploying an additional managed KIE Server for an authoring or managed environment".

For instructions about deploying immutable KIE Servers, see Section 9.1, "Deploying an immutable KIE Server using an S2I build" and Section 9.2, "Deploying an immutable KIE Server from KJAR services" .

8.1. DEPLOYING AN AUTHORING ENVIRONMENT

You can use OpenShift templates to deploy a single or high-availability authoring environment. This environment consists of Business Central and a single KIE Server.

8.1.1. Starting configuration of the template for an authoring environment

If you want to deploy a single authoring environment, use the **rhdm79-authoring.yaml** template file.

If you want to deploy a high-availability authoring environment, use the **rhdm79-authoring-ha.yaml** template file.

Procedure

1. Download the **rhdm-7.9.1-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
     DECISION_CENTRAL_HTTPS_SECRET=decisioncentral-app-secret -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 8.1.2, "Setting required parameters for an authoring environment" to set common parameters. You can view the template file to see descriptions for all parameters.

8.1.2. Setting required parameters for an authoring environment

When configuring the template to deploy an authoring environment, you must set the following parameters in all cases.

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, "Starting configuration of the template for an authoring environment".

Procedure

1. Set the following parameters:
   - **Credentials secret** (**CREDENTIALS_SECRET**): The name of the secret containing the administrative user credentials, as created in Section 6.5, "Creating the secret for the administrative user".
   - **Business Central Server Keystore Secret Name** (**DECISION_CENTRAL_HTTPS_SECRET**): The name of the secret for Business Central, as created in Section 6.3, "Creating the secrets for Business Central" .
   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for KIE Server, as created in Section 6.2, "Creating the secrets for KIE Server" .
   - **Business Central Server Certificate Name**(**DECISION_CENTRAL_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 6.3, "Creating the secrets for Business Central".
   - **Business Central Server Keystore Password** (**DECISION_CENTRAL_HTTPS_PASSWORD**): The password for the keystore that you created in Section 6.3, "Creating the secrets for Business Central" .
   - **KIE Server Certificate Name**(**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 6.2, "Creating the secrets for KIE Server" .
   - **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 6.2, "Creating the secrets for KIE Server" .
   - **Application Name** (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and KIE Server. OpenShift uses the application name to create a separate set of deployment configurations, services, routes, labels, and artifacts.
   - **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 6.1, "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.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 8.1.12, “Completing deployment of the template for an authoring environment”.

8.1.3. Configuring the image stream namespace for an authoring environment

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 8.1.1, “Starting configuration of the template for an authoring environment”.

Procedure

If you installed an image streams file according to instructions in Section 6.1, “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.

8.1.4. Setting an optional Maven repository for an authoring environment

When configuring the template to deploy an authoring environment, if you want to place the built KJAR files into an external Maven repository, you must set parameters to access the repository.

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, “Starting configuration of the template for an authoring environment”.

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 user name 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 8.1.12, “Completing deployment of the template for an authoring environment”.

IMPORTANT

To export or push Business Central projects as KJAR artifacts to the external Maven repository, you must also add the repository information in the **pom.xml** file for every project. For information about exporting Business Central projects to an external repository, see *Packaging and deploying a Red Hat Decision Manager project* .

8.1.5. Configuring access to a Maven mirror in an environment without a connection to the public Internet for an authoring environment

When configuring the template to deploy an authoring environment, 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 6.9, “Preparing a Maven mirror repository for offline use” .

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, “Starting configuration of the template for an authoring environment”.

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 6.9, “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:*,!repo-rhdmcentr**; with this value, Maven retrieves artifacts from the built-in Maven repository of Business Central directly and retrieves any other required artifacts from the mirror. If you configure an external Maven repository (**MAVEN_REPO_URL**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository, for example, **external:*,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**. 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 (**DECISION_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 8.1.12, “Completing deployment of the template for an authoring environment”.

8.1.6. Configuring Business Central and KIE Server replicas for a high-availability authoring environment

If you are deploying a high-availability authoring environment, by default two replicas of Business Central and two replicas of the KIE Server are initially created.

Optionally, you can modify the number of replicas.

Skip this procedure for a single authoring environment.

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, “Starting configuration of the template for an authoring environment”.

Procedure

To modify the numbers of initial replicas, set the following parameters:

- **Business Central Container Replicas**(**DECISION_CENTRAL_CONTAINER_REPLICAS**): The number of replicas that the deployment initially creates for Business Central.

- **KIE Server Container Replicas**(**KIE_SERVER_CONTAINER_REPLICAS**): The number of replicas that the deployment initially creates for the KIE Server.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 8.1.12, “Completing deployment of the template for an authoring environment”.

8.1.7. Specifying the Git hooks directory for an authoring environment

You can use Git hooks to facilitate interaction between the internal Git repository of Business Central and an external Git repository.

If you want to use Git hooks, you must configure a Git hooks directory.

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, “Starting configuration of the template for an authoring environment”.

Procedure

To configure a Git hooks directory, set the following parameter:

- **Git hooks directory** (**GIT_HOOKS_DIR**): The fully qualified path to a Git hooks directory, for example, **/opt/kie/data/git/hooks**. You must provide the content of this directory and mount it at the specified path. For instructions about providing and mounting the Git hooks directory using a configuration map or a persistent volume, see Section 10.1, “(Optional) Providing the Git hooks directory”.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 8.1.12, "Completing deployment of the template for an authoring environment".

8.1.8. Configuring resource usage for a high-availability deployment

If you are deploying the high-availability template (**rhdm79-authoring-ha.yaml**), you can optionally configure resource usage to optimize performance for your requirements.

If you are deploying the single authoring environment template (**rhdm79-authoring.yaml**), skip this procedure.

For more information about sizing resources, see the following sections in the Red Hat OpenShift Container Platform 3.11 product documentation:

- Application memory sizing
- Compute resources

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, "Starting configuration of the template for an authoring environment".

Procedure

Set the following parameters of the template as applicable:

- Business Central Container Memory Limit(**DECISION_CENTRAL_MEMORY_LIMIT**): The amount of memory requested in the OpenShift environment for the Business Central container. The default value is **8Gi**.
- Business Central JVM Max Memory Ratio (**DECISION_CENTRAL_JAVA_MAX_MEM_RATIO**): The percentage of container memory that is used for the Java Virtual Machine for Business Central. The remaining memory is used for the operating system. The default value is **80**, for a limit of 80%.
- Business Central Container CPU Limit(**DECISION_CENTRAL_CPU_LIMIT**): The maximum CPU usage for Business Central. The default value is **2000m**.
- KIE Server Container Memory Limit(**KIE_SERVER_MEMORY_LIMIT**): The amount of memory requested in the OpenShift environment for the KIE Server container. The default value is **1Gi**.
- KIE Server Container CPU Limit(**KIE_SERVER_CPU_LIMIT**): The maximum CPU usage for KIE Server. The default value is **1000m**.
- DataGrid Container Memory Limit(**DATAGRID_MEMORY_LIMIT**): The amount of memory requested in the OpenShift environment for the Red Hat Data Grid container. The default value is **2Gi**.
- DataGrid Container CPU Limit(**DATAGRID_CPU_LIMIT**): The maximum CPU usage for Red Hat Data Grid. The default value is **1000m**.

8.1.9. Setting parameters for RH-SSO authentication for an authoring environment

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy an authoring environment.

> **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 11, *Red Hat Decision Manager roles and users*.
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, “Creating the secret for the administrative user”. This user must have the **kie-server,rest-all,admin** roles.
- 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 8.1.1, “Starting configuration of the template for an authoring environment”.

Procedure

1. 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.
2. Complete one of the following procedures:
   a. If you created the clients 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.
      - **Business Central RH-SSO Client Secret**(**DECISION_CENTRAL_SSO_SECRET**): The secret string that is set in RH-SSO for the client for Business Central.
      - **KIE Server RH-SSO Client name**(**KIE_SERVER_SSO_CLIENT**): The RH-SSO client name for KIE Server.

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

b. To create the clients 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 name of the client to create in RH-SSO for Business Central.

   - **Business Central RH-SSO Client Secret**(**DECISION_CENTRAL_SSO_SECRET**): The secret string to set in RH-SSO for the client for Business Central.

   - **KIE Server RH-SSO Client name**(**KIE_SERVER_SSO_CLIENT**): The name of the client to create in RH-SSO for KIE Server.

   - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for KIE 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 8.1.12, "Completing deployment of the template for an authoring environment".

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.

8.1.10. Setting parameters for LDAP authentication for an authoring environment

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an authoring environment.

> **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 11, *Red Hat Decision Manager roles and users* .
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.

- You started the configuration of the template, as described in Section 8.1.1, "Starting configuration of the template for an authoring environment".

Procedure

1. 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 .

    NOTE

    If you want to enable LDAP failover, you can put set or more LDAP server addresses in the **AUTH_LDAP_URL** parameter, separated by a space.

    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 10.3, "(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 8.1.12, "Completing deployment of the template for an authoring environment".

8.1.11. Enabling Prometheus metric collection for an authoring environment

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

Prerequisites

- You started the configuration of the template, as described in Section 8.1.1, "Starting configuration of the template for an authoring environment".

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 8.1.12, "Completing deployment of the template for an authoring environment".

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

8.1.12. Completing deployment of the template for an authoring environment

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.

Next steps

Depending on your needs for the environment, optionally complete procedures described in Chapter 10, *Optional procedures after deploying your environment* .

8.2. ENABLING THE OPENSHIFTSTARTUPSTRATEGY SETTING TO CONNECT ADDITIONAL KIE SERVERS TO BUSINESS CENTRAL

In an environment deployed using Red Hat Decision Manager authoring templates, Business Central manages one KIE Server. You can scale the KIE Server pod, but all the copies execute the same services.

You can connect additional KIE Servers to Business Central. However, if you deployed a single authoring environment using the **rhdm79-authoring.yaml**, you must enable the **OpenShiftStartupStrategy** setting in the environment. When **OpenShiftStartupStrategy** is enabled, Business Central automatically discovers KIE Servers in the same namespace and these KIE Servers can be configured to connect to the Business Central.

With the **OpenShiftStartupStrategy** setting, when a user deploys a service to the KIE Server, the KIE Server deployment is rolled out again. Users can not deploy another service to the same KIE Server until the roll-out completes. Because the roll-out might take noticeable time, the **OpenShiftStartupStrategy** setting might not be suitable for some authoring environments.

Do not complete this procedure if you deployed a high-availability authoring environment using the **rhdm79-authoring-ha.yaml** template. In this environment, the **OpenShiftStartupStrategy** setting is enabled by default.

Do not complete this procedure unless you want to connect additional KIE Servers to Business Central.

Prerequisites

- You deployed an authoring environment using the **rhdm79-authoring.yaml** template.
- You are logged in to the OpenShift project where the environment is deployed using the **oc** tool.

Procedure

1. Enter the following command to view the deployment configurations that are deployed in the project:

```
$ oc get dc
```

2. In the output of the command, find the deployment configuration names for the Business Central and KIE Server pods:

    - The name of the deployment configuration for Business Central is ***myapp*-rhdmcentr**. Replace ***myapp*** with the application name of the environment, which is set in the **APPLICATION_NAME** parameter of the template.

    - The name of the deployment configuration for KIE Server is ***myapp*-kieserver**. Replace ***myapp*** with the application name.

3. Enter the following commands to enable the **OpenShiftStartupStrategy** setting on the pods:

    ```
    $ oc env myapp-rhdmcentr KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED=true
    $ oc env myapp-kieserver KIE_SERVER_STARTUP_STRATEGY=OpenShiftStartupStrategy
    ```

    In these commands, replace ***myapp*-rhdmcentr** with the Business Central deployment configuration name and ***myapp*-kieserver** with the KIE Server deployment configuration name.

4. When you enable the **OpenShiftStartupStrategy** setting, by default Business Central discovers only KIE Servers that are deployed with the same value of the **APPLICATION_NAME** parameter as the authoring template. If you want to connect KIE Servers with any other application names to the Business Central, enter the following command:

    ```
    $ oc env myapp-rhdmcentr
    KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED=true
    ```

    In this command, replace ***myapp*-rhdmcentr** with the Business Central deployment configuration name.

8.3. DEPLOYING AN ADDITIONAL MANAGED KIE SERVER FOR AN AUTHORING OR MANAGED ENVIRONMENT

You can deploy an additional managed KIE Server to an authoring or managed environment. Deploy the server in the same project as the Business Central deployment.

If you deployed a single authoring environment using the **rhdm79-authoring.yaml** template, you must enable the **OpenShiftStartupStrategy** setting for your environment for the Business Central to connect to the KIE Server. For instructions about enabling the **OpenShiftStartupStrategy** setting, see Section 8.2, “Enabling the **OpenShiftStartupStrategy** setting to connect additional KIE Servers to Business Central”. You do not need to complete this procedure for a high-availability authoring environment.

The KIE Server loads services from a Maven repository. You must configure the server to use either the Business Central built-in repository or an external repository.

The server starts with no loaded services. Use Business Central or the REST API of the KIE Server to deploy and undeploy services on the server.

8.3.1. Starting configuration of the template for an additional managed KIE Server

To deploy an additional managed KIE Server, use the **rhdm79-kieserver.yaml** template file.

Procedure

1. Download the **rhdm-7.9.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
2. Extract the **rhdm79-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 **rhdm79-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>/rhdm79-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 8.3.2, “Setting required parameters for an additional managed KIE Server” to set common parameters. You can view the template file to see descriptions for all parameters.

8.3.2. Setting required parameters for an additional managed KIE Server

When configuring the template to deploy an additional managed KIE Server, you must set the following parameters in all cases.

Prerequisites

- You started the configuration of the template, as described in Section 8.3.1, “Starting configuration of the template for an additional managed KIE Server”.

Procedure

1. Set the following parameters:
   - **Credentials secret** (**CREDENTIALS_SECRET**): The name of the secret containing the administrative user credentials, as created in Section 6.5, “Creating the secret for the administrative user”.
   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for KIE Server, as created in Section 6.2, “Creating the secrets for KIE Server” .
   - **KIE Server Certificate Name**(**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 6.2, “Creating the secrets for KIE Server” .
   - **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 6.2, “Creating the secrets for KIE Server” .

- Application Name (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and KIE 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 KIE Server joins on Business Central. If you are deploying several KIE Servers, you must ensure each of the servers has a different application name.

- KIE Server Mode(**KIE_SERVER_MODE**): In the **rhdm79-kieserver.yaml** template the default value is **PRODUCTION**. In **PRODUCTION** mode, you cannot deploy **SNAPSHOT** versions of KJAR artifacts on the KIE 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 KIE 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 6.1, "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.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 8.3.9, "Completing deployment of the template for an additional managed KIE Server".

8.3.3. Configuring the image stream namespace for an additional managed KIE Server

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 8.3.1, "Starting configuration of the template for an additional managed KIE Server".

Procedure

If you installed an image streams file according to instructions in Section 6.1, "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.

8.3.4. Configuring information about a Business Central instance for an additional managed KIE Server

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

The Business Central instance must be configured with the same credentials secret (**CREDENTIALS_SECRET**) as the KIE Server.

Prerequisites

- You started the configuration of the template, as described in Section 8.3.1, "Starting configuration of the template for an additional managed KIE Server".

Procedure

1. Set the following parameters:
    - **Name of the Business Central service**(**DECISION_CENTRAL_SERVICE**): The OpenShift service name for the Business Central.
2. Configure access to the Maven repository from which the server must load services. You must configure the same repository that the Business Central uses.
    - If the Business Central uses its own built-in repository, set the following parameter:
        - **Name of the Maven service hosted by Business Central** (**DECISION_CENTRAL_MAVEN_SERVICE**): The OpenShift service name for the Business Central.
    - If you configured the Business Central to use an external Maven repository, set the following parameters:
        - **Maven repository URL** (**MAVEN_REPO_URL**): A URL for the external Maven repository that Business Central uses.
        - **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 user name 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 8.3.9, "Completing deployment of the template for an additional managed KIE Server".

8.3.5. Configuring access to a Maven mirror in an environment without a connection to the public Internet for an additional managed KIE Server

When configuring the template to deploy an additional managed KIE Server, 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 6.9, "Preparing a Maven mirror repository for offline use" .

Prerequisites

- You started the configuration of the template, as described in Section 8.3.1, "Starting configuration of the template for an additional managed KIE Server".

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 6.9, "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 (**DECISION_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 8.3.9, "Completing deployment of the template for an additional managed KIE Server".

8.3.6. Setting parameters for RH-SSO authentication for an additional managed KIE Server

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy an additional managed KIE Server.

> **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 11, *Red Hat Decision Manager roles and users*.
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.
- 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 8.3.1, "Starting configuration of the template for an additional managed KIE Server".

Procedure

1. 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.
2. 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 KIE Server.
      - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string that is set in RH-SSO for the client for KIE 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 KIE Server.
      - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for KIE 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 8.3.9, "Completing deployment of the template for an additional managed KIE Server".

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.

8.3.7. Setting parameters for LDAP authentication for an additional managed KIE Server

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an additional managed KIE Server.

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 11, *Red Hat Decision Manager roles and users* .
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.
- You started the configuration of the template, as described in Section 8.3.1, "Starting configuration of the template for an additional managed KIE Server".

Procedure

1. 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 .

   NOTE

   If you want to enable LDAP failover, you can put set or more LDAP server addresses in the **AUTH_LDAP_URL** parameter, separated by a space.

   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 10.3, "(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 8.3.9, "Completing deployment of the template for an additional managed KIE Server".

8.3.8. Enabling Prometheus metric collection for an additional managed KIE Server

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

Prerequisites

- You started the configuration of the template, as described in Section 8.3.1, “Starting configuration of the template for an additional managed KIE Server”.

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 8.3.9, “Completing deployment of the template for an additional managed KIE Server”.

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

8.3.9. Completing deployment of the template for an additional managed KIE Server

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.

Next steps

Depending on your needs for the environment, optionally complete procedures described in Chapter 10, *Optional procedures after deploying your environment* .

CHAPTER 9. ENVIRONMENT WITH IMMUTABLE SERVERS

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

On an immutable KIE 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 KIE Server. The advantage of this approach is that the KIE Server with the services in it runs like any other containerized service and does not require specialized management. The KIE Server runs like any other pod on the OpenShift environment; you can use any container-based integration workflows as necessary.

When you create a KIE 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 KIE 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 KIE 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.

9.1. DEPLOYING AN IMMUTABLE KIE SERVER USING AN S2I BUILD

You can deploy an immutable KIE 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 KIE Server. You can use Business Central to view monitoring information. The KIE 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 KIE 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 KIE Server automatically. You can use Business Central to start and stop (but not deploy) services on the immutable KIE Server.

9.1.1. Starting configuration of the template for an immutable KIE Server using S2I

To deploy an immutable KIE Server using an S2I build, use the **rhdm79-prod-immutable-kieserver-amq.yaml** template file if you want to enable JMS capabilities. Otherwise, use the **rhdm79-prod-immutable-kieserver.yaml** template file.

Procedure

1. Download the **rhdm-7.9.1-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 9.1.2, "Setting required parameters for an immutable KIE Server using S2I" to set common parameters. You can view the template file to see descriptions for all parameters.

9.1.2. Setting required parameters for an immutable KIE Server using S2I

When configuring the template to deploy an immutable KIE 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 9.1.1, "Starting configuration of the template for an immutable KIE Server using S2I".

Procedure

1. Set the following parameters:

   - **Credentials secret** (**CREDENTIALS_SECRET**): The name of the secret containing the administrative user credentials, as created in Section 6.5, "Creating the secret for the administrative user".

   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for KIE Server, as created in Section 6.2, "Creating the secrets for KIE Server" .

   - **KIE Server Certificate Name**(**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 6.2, "Creating the secrets for KIE Server" .

- KIE Server Keystore Password (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 6.2, "Creating the secrets for KIE Server".

- Application Name (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and KIE 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 KIE Server joins on Business Central. If you are deploying several KIE 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 6.1, "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.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 9.1.11, "Completing deployment of the template for an immutable KIE Server using S2I".

9.1.3. Configuring the image stream namespace for an immutable KIE 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 9.1.1, “Starting configuration of the template for an immutable KIE Server using S2I”.

Procedure

If you installed an image streams file according to instructions in Section 6.1, “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.

9.1.4. Configuring information about a Business Central instance for an immutable KIE Server using S2I

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

The Business Central instance must be configured with the same credentials secret (**CREDENTIALS_SECRET**) as the KIE Server.

Prerequisites

- You started the configuration of the template, as described in Section 9.1.1, “Starting configuration of the template for an immutable KIE Server using S2I”.

Procedure

1. Set the following parameters:
    - **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 9.1.11, “Completing deployment of the template for an immutable KIE Server using S2I”.

9.1.5. Setting an optional Maven repository for an immutable KIE Server using S2I

When configuring the template to deploy an immutable KIE 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 9.1.1, “Starting configuration of the template for an immutable KIE 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 user name 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 9.1.11, “Completing deployment of the template for an immutable KIE Server using S2I”.

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

When configuring the template to deploy an immutable KIE 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 6.9, “Preparing a Maven mirror repository for offline use” .

Prerequisites

- You started the configuration of the template, as described in Section 9.1.1, “Starting configuration of the template for an immutable KIE 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 6.9, “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 (**DECISION_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 9.1.11, “Completing deployment of the template for an immutable KIE Server using S2I”.

9.1.7. Configuring communication with an AMQ server for an immutable KIE Server using S2I

If you use the **rhdm79-prod-immutable-kieserver-amq.yaml** template file, JMS capabilities of the KIE 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 9.1.1, “Starting configuration of the template for an immutable KIE Server using S2I”, using the **rhdm79-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 KIE 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 9.1.11, “Completing deployment of the template for an immutable KIE Server using S2I”.

9.1.8. Setting parameters for RH-SSO authentication for an immutable KIE Server using S2I

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy an immutable KIE 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 11, *Red Hat Decision Manager roles and users*.
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.
- 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 9.1.1, "Starting configuration of the template for an immutable KIE Server using S2I".

Procedure

1. 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.
2. 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 KIE Server.
      - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string that is set in RH-SSO for the client for KIE 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 KIE Server.
  - **KIE Server RH-SSO Client Secret** (**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for KIE 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 9.1.11, "Completing deployment of the template for an immutable KIE 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.

9.1.9. Setting parameters for LDAP authentication for an immutable KIE Server using S2I

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an immutable KIE 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 11, *Red Hat Decision Manager roles and users* .
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.
- You started the configuration of the template, as described in Section 9.1.1, "Starting configuration of the template for an immutable KIE Server using S2I".

Procedure

1. 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 .

NOTE

If you want to enable LDAP failover, you can put set or more LDAP server addresses in the **AUTH_LDAP_URL** parameter, separated by a space.

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 10.3, "(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 9.1.11, "Completing deployment of the template for an immutable KIE Server using S2I".

9.1.10. Enabling Prometheus metric collection for an immutable KIE Server using S2I

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

Prerequisites

- You started the configuration of the template, as described in Section 9.1.1, "Starting configuration of the template for an immutable KIE 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 9.1.11, "Completing deployment of the template for an immutable KIE Server using S2I".

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

9.1.11. Completing deployment of the template for an immutable KIE 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.

Next steps

Depending on your needs for the environment, optionally complete procedures described in Chapter 10, *Optional procedures after deploying your environment* .

9.2. DEPLOYING AN IMMUTABLE KIE SERVER FROM KJAR SERVICES

You can deploy an immutable KIE 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 KIE Server. You can use Business Central to view monitoring information. The KIE 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 KIE Server automatically. You can use Business Central to start and stop (but not deploy) services on the immutable KIE Server and to view monitoring data.

9.2.1. Starting configuration of the template for an immutable KIE Server from KJAR services

To deploy an immutable KIE Server from KJAR services, use the **rhdm79-kieserver.yaml** template file.

Procedure

1. Download the **rhdm-7.9.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
2. Extract the **rhdm79-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 **rhdm79-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>/rhdm79-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 9.2.2, "Setting required parameters for an immutable KIE Server from KJAR services" to set common parameters. You can view the template file to see descriptions for all parameters.

9.2.2. Setting required parameters for an immutable KIE Server from KJAR services

When configuring the template to deploy an immutable KIE 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 9.2.1, "Starting configuration of the template for an immutable KIE Server from KJAR services".

Procedure

1. Set the following parameters:
   - **Credentials secret** (**CREDENTIALS_SECRET**): The name of the secret containing the administrative user credentials, as created in Section 6.5, "Creating the secret for the administrative user".
   - **KIE Server Keystore Secret Name**(**KIE_SERVER_HTTPS_SECRET**): The name of the secret for KIE Server, as created in Section 6.2, "Creating the secrets for KIE Server" .
   - **KIE Server Certificate Name** (**KIE_SERVER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 6.2, "Creating the secrets for KIE Server" .
   - **KIE Server Keystore Password** (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 6.2, "Creating the secrets for KIE Server" .
   - **Application Name** (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central Monitoring and KIE 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 KIE Server joins on Business Central. If you are deploying several KIE 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 KIE 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 user name 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 **rhdm79-kieserver-*.yaml** templates the default value is **PRODUCTION**. In **PRODUCTION** mode, you cannot deploy **SNAPSHOT** versions of KJAR artifacts on the KIE 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 KIE 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 6.1, "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.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 9.2.9, "Completing deployment of the template for an immutable KIE Server from KJAR services".

9.2.3. Configuring the image stream namespace for an immutable KIE 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 9.2.1, "Starting configuration of the template for an immutable KIE Server from KJAR services".

Procedure

If you installed an image streams file according to instructions in Section 6.1, "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.

9.2.4. Configuring information about a Business Central instance for an immutable KIE Server from KJAR services

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

The Business Central instance must be configured with the same credentials secret (**CREDENTIALS_SECRET**) as the KIE Server.

Prerequisites

- You started the configuration of the template, as described in Section 9.2.1, "Starting configuration of the template for an immutable KIE Server from KJAR services".

Procedure

1. Set the following parameters:
   - **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 from which services must be deployed.
   - **Maven repository username** (**MAVEN_REPO_USERNAME**): The user name 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 9.2.9, "Completing deployment of the template for an immutable KIE Server from KJAR services".

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

When configuring the template to deploy an immutable KIE 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 6.9, "Preparing a Maven mirror repository for offline use".

Prerequisites

- You started the configuration of the template, as described in Section 9.2.1, "Starting configuration of the template for an immutable KIE 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 6.9, "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 (**DECISION_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 9.2.9, “Completing deployment of the template for an immutable KIE Server from KJAR services”.

9.2.6. Setting parameters for RH-SSO authentication for an immutable KIE 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 KIE 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 11, *Red Hat Decision Manager roles and users*.
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, “Creating the secret for the administrative user”. This user must have the **kie-server,rest-all,admin** roles.
- 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 9.2.1, “Starting configuration of the template for an immutable KIE Server from KJAR services”.

Procedure

1. 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.
2. 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 KIE Server.
      - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string that is set in RH-SSO for the client for KIE 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 KIE Server.
      - **KIE Server RH-SSO Client Secret**(**KIE_SERVER_SSO_SECRET**): The secret string to set in RH-SSO for the client for KIE 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 9.2.9, “Completing deployment of the template for an immutable KIE 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.

9.2.7. Setting parameters for LDAP authentication for an immutable KIE Server from KJAR services

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy an immutable KIE 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 11, *Red Hat Decision Manager roles and users* .
  You must create a user with the username and password configured in the secret for the administrative user, as described in Section 6.5, "Creating the secret for the administrative user". This user must have the **kie-server,rest-all,admin** roles.

- You started the configuration of the template, as described in Section 9.2.1, "Starting configuration of the template for an immutable KIE Server from KJAR services".

Procedure

1. 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 .

   NOTE

   If you want to enable LDAP failover, you can put set or more LDAP server addresses in the **AUTH_LDAP_URL** parameter, separated by a space.

   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 10.3, "(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 9.2.9, "Completing deployment of the template for an immutable KIE Server from KJAR services".

9.2.8. Enabling Prometheus metric collection for an immutable KIE Server from KJAR services

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

Prerequisites

- You started the configuration of the template, as described in Section 9.2.1, “Starting configuration of the template for an immutable KIE 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 9.2.9, “Completing deployment of the template for an immutable KIE Server from KJAR services”.

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

9.2.9. Completing deployment of the template for an immutable KIE 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.

Next steps

Depending on your needs for the environment, optionally complete procedures described in Chapter 10, *Optional procedures after deploying your environment* .

CHAPTER 10. OPTIONAL PROCEDURES AFTER DEPLOYING YOUR ENVIRONMENT

Depending on the needs for your environment, you might need to complete certain optional procedures after deploying it.

10.1. (OPTIONAL) PROVIDING THE GIT HOOKS DIRECTORY

If you deploy an authoring enviropnent and configure the **GIT_HOOKS_DIR** parameter, you must provide a directory of Git hooks and must mount this directory on the Business Central deployment.

The typical use of Git hooks is interaction with an upstream repository. To enable Git hooks to push commits into an upstream repository, you must also provide a secret key that corresponds to a public key configured on the upstream repository.

Prerequisites

- You deployed a Red Hat Decision Manager authoring environment using templates
- You set the **GIT_HOOKS_DIR** parameter in the deployment

Procedure

1. If interaction with an upstream repository using SSH authentication is required, complete the following steps to prepare and mount a secret with the necessary files:
    a. Prepare the **id_rsa** file with a private key that matches a public key stored in the repository.
    b. Prepare the **known_hosts** file with the correct name, address, and public key for the repository.
    c. Create a secret with the two files using the **oc** command, for example:

    ```
    oc create secret git-hooks-secret --from-file=id_rsa=id_rsa --from-file=known_hosts=known_hosts
    ```

    d. Mount the secret in the SSH key path of the Business Central deployment, for example:

    ```
    oc set volume dc/<myapp>-rhdmcentr --add --type secret --secret-name git-hooks-secret
    --mount-path=/home/jboss/.ssh --name=ssh-key
    ```

    Replace **<myapp>** with the application name that you set when configuring the template.

2. Create the Git hooks directory. For instructions, see the Git hooks reference documentation. For example, a simple Git hooks directory can provide a post-commit hook that pushes the changes upstream. If the project was imported into Business Central from a repository, this repository remains configured as the upstream repository. Create a file named **post-commit** with permission values **755** and the following content:

    ```
    git push
    ```

NOTE

A **pre-commit** script is not supported in Business Central. Use a **post-commit** script.

3. Supply the Git hooks directory to the Business Central deployment. You can use a configuration map or a persistent volume.

   a. If the Git hooks consist of one or several fixed script files, use a configuration map. Complete the following steps:

      i. Change into the Git hooks directory that you have created.

      ii. Create an OpenShift configuration map from the files in the directory. Run the following command:

      ```
      oc create configmap git-hooks --from-file=<file_1>=<file_1> --from-file=<file_2>=
      <file_2> ...
      ```

      Replace **file_1**, **file_2**, and so on with Git hook script file names. Example:

      ```
      oc create configmap git-hooks --from-file=post-commit=post-commit
      ```

      iii. Mount the configuration map on the Business Central deployment in the path that you have configured:

      ```
      oc set volume dc/<myapp>-rhdmcentr --add --type configmap --configmap-name git-
      hooks  --mount-path=<git_hooks_dir> --name=git-hooks
      ```

      Replace **<myapp>** with the application name that was set when configuring the template and **<git_hooks_dir>** is the value of **GIT_HOOKS_DIR** that was set when configuring the template.

   b. If the Git hooks consist of long files or depend on binaries, such as executable or KJAR files, use a persistence volume. You must create a persistent volume, create a persistent volume claim and associate the volume with the claim, transfer files to the volume, and mount the volume in the ***myapp*-rhdmcentr** deployment configuration (replace *myapp* with the application name). For instructions about creating and mounting persistence volumes, see Using persistent volumes. For instructions about copying files onto a persistent volume, see Transferring files in and out of containers .

4. Wait a few minutes, then review the list and status of pods in your project. Because Business Central does not start until you provide the Git hooks directory, the KIE Server might not start at all. To see if it has started, check the output of the following command:

   ```
   oc get pods
   ```

   If a working KIE Server pod is not present, start it:

   ```
   oc rollout latest dc/<myapp>-kieserver
   ```

   Replace **<myapp>** with the application name that was set when configuring the template.

10.2. (OPTIONAL) PROVIDING A TRUSTSTORE FOR ACCESSING HTTPS SERVERS WITH SELF-SIGNED CERTIFICATES

Components of your Red Hat Decision Manager infrastructure might need to use HTTPS access to servers that have a self-signed HTTPS certificate. For example, Business Central and KIE Server might need to interact with an internal Nexus repository that uses a self-signed HTTPS server certificate.

In this case, to ensure that HTTPS connections complete successfully, you must provide client certificates for these services using a truststore.

Skip this procedure if you do not need Red Hat Decision Manager components to communicate with servers that use self-signed HTTPS server certificates.

Prerequisites

- You deployed a Red Hat Decision Manager environment using templates
- You have the client certificates that you want to add to the deployment

Procedure

1. Prepare a truststore with the certificates. Use the following command to create a truststore or to add a certificate to an existing truststore. Add all the necessary certificates to one truststore.

   ```
   keytool -importcert -file certificate-file -alias alias -keyalg algorithm -keysize size -trustcacerts -noprompt -storetype JKS -keypass truststore-password -storepass truststore-password -keystore keystore-file
   ```

   Replace the following values:

   - ***certificate-file***: The pathname of the certificate that you want to add to the truststore.
   - ***alias***: The alias for the certificate in the truststore. If you are adding more than one certificate to the truststore, every certificate must have a unique alias.
   - ***algorithm***: The encryption algorithm used for the certificate, typically **RSA**.
   - ***size***: The size of the certificate key in bytes, for example, **2048**.
   - ***truststore-password***: The password for the truststore.
   - ***keystore-file***: The pathname of the truststore file. If the file does not exist, the command creates a new truststore.
     The following example command adds a certificate from the **/var/certs/nexus.cer** file to a truststore in the **/var/keystores/custom-trustore.jks** file. The truststore password is **mykeystorepass**.

     ```
     keytool -importcert -file /var/certs/nexus.cer -alias nexus-cert -keyalg RSA -keysize 2048 -trustcacerts -noprompt -storetype JKS -keypass mykeystorepass -storepass mykeystorepass -keystore /var/keystores/custom-trustore.jks
     ```

2. Create a secret with the truststore file using the **oc** command, for example:

   ```
   oc create secret generic truststore-secret --from-file=/var/keystores/custom-trustore.jks
   ```

3. In the deployment for the necessary components of your infrastructure, mount the secret and then set the **JAVA_OPTS_APPEND** option to enable the Java application infrastructure to use the trast store, for example:

   ```
   oc set volume dc/myapp-rhdmcentr --add --overwrite --name=custom-trustore-volume --
   mount-path /etc/custom-secret-volume --secret-name=custom-secret

   oc set env dc/myapp-rhdmcentr JAVA_OPTS_APPEND='-
   Djavax.net.ssl.trustStore=/etc/custom-secret-volume/custom-trustore.jks -
   Djavax.net.ssl.trustStoreType=jks -Djavax.net.ssl.trustStorePassword=mykeystorepass'
   ```

   ```
   oc set volume dc/myapp-kieserver --add --overwrite --name=custom-trustore-volume --
   mount-path /etc/custom-secret-volume --secret-name=custom-secret

   oc set env dc/myapp-kieserver JAVA_OPTS_APPEND='-
   Djavax.net.ssl.trustStore=/etc/custom-secret-volume/custom-trustore.jks -
   Djavax.net.ssl.trustStoreType=jks -Djavax.net.ssl.trustStorePassword=mykeystorepass'
   ```

   Replace ***myapp*** with the application name that you set when configuring the template.

10.3. (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.

Prerequisites

- You deployed a Red Hat Decision Manager environment using templates
- You set the **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** parameter in the deployment

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:

   Replace **myapp** with the application name. Sometimes, several KIE 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 11. RED HAT DECISION MANAGER ROLES AND USERS

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

The Business Central and KIE Server use Java Authentication and Authorization Service (JAAS) login module to authenticate the users. If both Business Central and KIE 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 KIE Server.

However, if Business Central and KIE 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 KIE Server (for example, to view or manage process definitions in Business Central). In case, the user is not authenticated on the KIE 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 KIE Server. For this reason, the available roles can differ depending on whether Business Central, KIE 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 KIE Server (KIE Server) REST capabilities.

CHAPTER 12. OPENSHIFT TEMPLATE REFERENCE INFORMATION

Red Hat Decision Manager provides the following OpenShift templates. To access the templates, download and extract the **rhdm-7.9.1-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat customer portal.

- **rhdm79-trial-ephemeral.yaml** provides a Business Central and a KIE Server connected to the Business Central. This environment uses an ephemeral configuration without any persistent storage. For details about this template, see Section 12.1, “rhdm79-trial-ephemeral.yaml template”.
- **rhdm79-authoring.yaml** provides a Business Central and a KIE Server connected to the Business Central. You can use this environment to author services and other business assets or to run them in staging or production environments. For details about this template, see Section 12.2, “rhdm79-authoring.yaml template” .
- **rhdm79-authoring-ha.yaml** provides a high-availability Business Central and a KIE Server connected to the Business Central. You can use this environment to author services and other business assets or to run them in staging or production environments. For details about this template, see Section 12.3, “rhdm79-authoring-ha.yaml template” .
- **rhdm79-kieserver.yaml** provides a KIE Server. You can configure the KIE Server to connect to a Business Central. In this way, you can set up a staging or production environment in which one Business Central manages several distinct KIE Servers. For details about this template, see Section 12.4, “rhdm79-kieserver.yaml template” .
- **rhdm79-prod-immutable-kieserver.yaml** provides an immutable KIE Server. Deployment of this template includes a source-to-image (S2I) build for one or several services that are to run on the KIE Server. For details about this template, see Section 12.5, “rhdm79-prod-immutable-kieserver.yaml template”.
- **rhdm79-prod-immutable-kieserver-amq.yaml** provides an immutable KIE Server. Deployment of this template includes a source-to-image (S2I) build for one or several services that are to run on the KIE Server. This version of the template includes JMS integration. For details about this template, see Section 12.6, “rhdm79-prod-immutable-kieserver-amq.yaml template” .

12.1. RHDM79-TRIAL-EPHEMERAL.YAML TEMPLATE

Application template for an ephemeral authoring and testing environment, for Red Hat Decision Manager 7.9 - Deprecated

12.1.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
DEFAULT_PASSWORD	**KIE_ADMIN_PWD**	Default password used for multiple components for user convenience in this trial environment.	RedHat	True
KIE_ADMIN_USER	**KIE_ADMIN_USER**	KIE administrator user name.	adminUser	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_MODE	**KIE_SERVER_MODE**	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**DEVELOPMENT**	False

Variable name	Image Environment Variable	Description	Example value	Required
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
KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN	**AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE**	Sets the Access-Control-Allow-Origin response header value in the KIE Server (useful for CORS support).	*	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS	**AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE**	Sets the Access-Control-Allow-Methods response header value in the KIE Server (useful for CORS support).	GET, POST, OPTIONS, PUT	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS	**AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE**	Sets the Access-Control-Allow-Headers response header value in the KIE Server (useful for CORS support).	Accept, Authorization, Content-Type, X-Requested-With	False
KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS	**AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE**	Sets the Access-Control-Allow-Credentials response header value in the KIE Server (useful for CORS support).	true	False
KIE_SERVER_ACCESS_CONTROL_MAX_AGE	**AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE**	Sets the Access-Control-Max-Age response header value in the KIE Server (useful for CORS support).	1	False
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	–	False
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	false	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	true	False
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	60000	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 need to modify this parameter only 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 "7.9.0".	7.9.0	True
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	–	False
MAVEN_REPO_ID	**MAVEN_REPO_ID**	The id to use for the maven repository, if set. Default is generated randomly.	repo-custom	False
MAVEN_REPO_URL	**MAVEN_REPO_URL**	Fully qualified URL to a Maven repository or service.	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	User name for accessing the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Password to access the Maven repository, if required.	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	The directory to use for git hooks, if required.	**/opt/kie/data/git/hooks**	False

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central Container memory limit.	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	False
SSO_URL	**SSO_URL**	RH-SSO URL.	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name.	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO Client name.	–	False
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	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 user name 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 user name.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=example,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

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. 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 user name. 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 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.	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 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

12.1.2. 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.

12.1.2.1. Services

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

Service	Port	Name	Description
${APPLICATION_NAME}-rhdmcentr	8080	http	All the Decision Central web server's ports.
${APPLICATION_NAME}-kieserver	8080	–	All the KIE server web server's ports.

12.1.2.2. 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. See the Openshift documentation for more information.

Service	Security	Hostname
insecure-${APPLICATION_NAME}-rhdmcentr-http	none	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
insecure-${APPLICATION_NAME}-kieserver-http	none	**${KIE_SERVER_HOSTNAME_HTTP}**

12.1.2.3. 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. See the Openshift documentation for more information.

12.1.2.3.1. Triggers

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

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

12.1.2.3.2. 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. See the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-rhdmcentr	1
${APPLICATION_NAME}-kieserver	1

12.1.2.3.3. Pod Template

12.1.2.3.3.1. Service Accounts

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

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

12.1.2.3.3.2. Image

Deployment	Image
${APPLICATION_NAME}-rhdmcentr	rhdm-decisioncentral-rhel8
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.1.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

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

12.1.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

${APPLICATION_NAME}-kieserver

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

12.1.2.3.3.5. Exposed Ports

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-rhdmcentr	jolokia	8778	**TCP**
	http	8080	**TCP**

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

12.1.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-rhdmcentr	**KIE_ADMIN_USER**	KIE administrator user name.	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	Default password used for multiple components for user convenience in this trial environment.	**${DEFAULT_PASSWORD}**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	true
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	WORKBENCH_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-rhdmcentr
	MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	User name for accessing the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	GIT_HOOKS_DIR	The directory to use for git hooks, if required.	**${GIT_HOOKS_DIR}**
	KUBERNETES_NAMESPACE	–	–
	SSO_URL	RH-SSO URL.	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO Client Secret.	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO Client name.	**${DECISION_CENTRAL_SSO_CLIENT}**

Deployment	Variable name	Description	Example value
	SSO_USERNAME	RH-SSO Realm admin user name 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.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	KIE administrator user name.	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	Default password used for multiple components for user convenience in this trial environment.	**${DEFAULT_PASSWORD}**
	KIE_SERVER_MODE	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**${KIE_SERVER_MODE}**

Deployment	Variable name	Description	Example value
	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}**
	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	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL

Deployment	Variable name	Description	Example value
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	KIE administrator user name.	**${KIE_ADMIN_USER}**
	RHDMCENTR_MAVEN_REPO_PASSWORD	Default password used for multiple components for user convenience in this trial environment.	**${DEFAULT_PASSWORD}**
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	User name for accessing the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	SSO_URL	RH-SSO URL.	**${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}**

Deployment	Variable name	Description	Example value
	SSO_CLIENT	KIE Server RH-SSO Client name.	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name used to create the Client if it doesn't exist.	**${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 user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTP}**
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. This option is used together with usernameBeginString and usernameEndString.	**${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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**
	FILTERS	–	AC_ALLOW_ORIGIN,AC_ALLOW_METHODS,AC_ALLOW_HEADERS,AC_ALLOW_CREDENTIALS,AC_MAX_AGE
	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Origin
	AC_ALLOW_ORIGIN_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Origin response header value in the KIE Server (useful for CORS support).	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_ORIGIN}**
	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Methods
	AC_ALLOW_METHODS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Methods response header value in the KIE Server (useful for CORS support).	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_METHODS}**

Deployment	Variable name	Description	Example value
	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Headers
	AC_ALLOW_HEADERS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Headers response header value in the KIE Server (useful for CORS support).	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_HEADERS}**
	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Allow-Credentials
	AC_ALLOW_CREDENTIALS_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Allow-Credentials response header value in the KIE Server (useful for CORS support).	**${KIE_SERVER_ACCESS_CONTROL_ALLOW_CREDENTIALS}**
	AC_MAX_AGE_FILTER_RESPONSE_HEADER_NAME	–	Access-Control-Max-Age
	AC_MAX_AGE_FILTER_RESPONSE_HEADER_VALUE	Sets the Access-Control-Max-Age response header value in the KIE Server (useful for CORS support).	**${KIE_SERVER_ACCESS_CONTROL_MAX_AGE}**
	KUBERNETES_NAMESPACE	–	–

12.1.2.4. External Dependencies

12.1.2.4.1. Secrets

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

12.2. RHDM79-AUTHORING.YAML TEMPLATE

Application template for a non-HA persistent authoring environment, for Red Hat Decision Manager 7.9 - Deprecated

12.2.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
CREDENTIALS_SECRET	–	Secret containing the KIE_ADMIN_USER and KIE_ADMIN_PWD values.	rhpam-credentials	True
KIE_SERVER_CONTROLLER_TOKEN	**KIE_SERVER_CONTROLLER_TOKEN**	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	–	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

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_MODE	**KIE_SERVER_MODE**	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**DEVELOPMENT**	False
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

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route for Decision Central. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	–	False
DECISION_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route for Decision Central. Leave blank for default hostname, e.g.: <application-name>-rhdmcentr-<project>.<default-domain-suffix>	–	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route for KIE Server. 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 for KIE Server. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_HTTPS_SECRET	–	The name of the secret containing the keystore file for Decision Central.	decisioncentral-app-secret	True
DECISION_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret.	keystore.jks	False
DECISION_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate.	jboss	False
DECISION_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate.	mykeystorepass	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_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	false	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	true	False
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	60000	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 need to modify this parameter only 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 "7.9.0".	7.9.0	True
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Maven mirror that Decision Central and 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:*,!repo-rhdmcentr	False
MAVEN_REPO_ID	**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

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_REPO_URL	**MAVEN_REPO_URL**	Fully qualified URL to a Maven repository or service.	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	User name for accessing the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Password to access the Maven repository, if required.	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	The directory to use for git hooks, if required.	**/opt/kie/data/git/hooks**	False
DECISION_CENTRAL_VOLUME_CAPACITY	–	Size of the persistent storage for Decision Central's runtime data.	1Gi	True
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central Container memory limit.	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	False
SSO_URL	**SSO_URL**	RH-SSO URL.	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name.	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO Client name	–	False

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	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 user name 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 user name.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=example,ou=com	False

Variable name	Image Environment Variable	Description	Example value	Required
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
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	The timeout in milliseconds for user or role searches.	10000	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. 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 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.	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 roleNameAttributeId 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

12.2.2. 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.

12.2.2.1. Services

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

Service	Port	Name	Description
${APPLICATION_NAME}-rhdmcentr	8080	http	All the Decision Central web server's ports.
	8443	https	
${APPLICATION_NAME}-kieserver	8080	http	All the KIE server web server's ports.
	8443	https	

12.2.2.2. 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. See the Openshift documentation for more information.

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

12.2.2.3. 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. See the Openshift documentation for more information.

12.2.2.3.1. Triggers

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

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

12.2.2.3.2. 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. See the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-rhdmcentr	1
${APPLICATION_NAME}-kieserver	1

12.2.2.3.3. Pod Template

12.2.2.3.3.1. Service Accounts

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

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

12.2.2.3.3.2. Image

Deployment	Image
${APPLICATION_NAME}-rhdmcentr	rhdm-decisioncentral-rhel8
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.2.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

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

12.2.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

${APPLICATION_NAME}-kieserver

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

12.2.2.3.3.5. Exposed Ports

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

12.2.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-rhdmcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**

Deployment	Variable name	Description	Example value
	KIE_ADMIN_USER	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	false
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhdmcentr**
	MAVEN_MIRROR_URL	Maven mirror that Decision Central and KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	**${MAVEN_MIRROR_URL}**
	MAVEN_REPO_ID	The id to use for the maven repository. If set, it can be excluded from the optionally configured mirror by adding it to MAVEN_MIRROR_OF. For example: external:*,!repo-rhdmcentr,!repo-custom. If MAVEN_MIRROR_URL is set but MAVEN_MIRROR_ID is not set, an id will be generated randomly, but won't be usable in MAVEN_MIRROR_OF.	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	User name for accessing the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**

Deployment	Variable name	Description	Example value
	GIT_HOOKS_DIR	The directory to use for git hooks, if required.	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/decisioncentral-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret.	**${DECISION_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate.	**${DECISION_CENTRAL_HTTPS_NAME}**
	HTTPS_PASSWORD	The password for the keystore and certificate.	**${DECISION_CENTRAL_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–
	SSO_URL	RH-SSO URL.	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO Client Secret.	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO Client name	**${DECISION_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name 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.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route for Decision Central. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route for Decision Central. Leave blank for default hostname, e.g.: <application-name>-rhdmcentr-<project>.<default-domain-suffix>	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. This option is used together with usernameBeginString and usernameEndString.	**${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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_SERVER_MODE	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**${KIE_SERVER_MODE}**
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${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)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	ControllerBasedStartupStrategy
	MAVEN_MIRROR_URL	Maven mirror that Decision Central and KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	**${MAVEN_MIRROR_URL}**

Deployment	Variable name	Description	Example value
	MAVEN_MIRROR_OF	Maven mirror configuration for KIE server.	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	Set according to the credentials secret
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	Set according to the credentials secret
	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.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**

Deployment	Variable name	Description	Example value
	EXTERNAL_MAVEN_REPO_USERNAME	User name for accessing 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.	**${KIE_SERVER_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–
	SSO_URL	RH-SSO URL.	**${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.	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name used to create the Client if it doesn’t exist.	**${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.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route for KIE Server. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route for KIE Server. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**

12.2.2.3.3.7. Volumes

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

12.2.2.4. External Dependencies

12.2.2.4.1. Volume Claims

A **PersistentVolume** object is a storage resource in an OpenShift cluster. Storage is provisioned by an administrator by creating **PersistentVolume** objects from sources such as GCE Persistent Disks, AWS Elastic Block Stores (EBS), and NFS mounts. See the Openshift documentation for more information.

Name	Access Mode
${APPLICATION_NAME}-rhdmcentr-claim	ReadWriteOnce

12.2.2.4.2. Secrets

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

decisioncentral-app-secret kieserver-app-secret

12.3. RHDM79-AUTHORING-HA.YAML TEMPLATE

Application template for a HA persistent authoring environment, for Red Hat Decision Manager 7.9 - Deprecated

12.3.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
CREDENTIALS_SECRET	–	Secret containing the KIE_ADMIN_USER and KIE_ADMIN_PWD values.	rhpam-credentials	True
KIE_SERVER_CONTROLLER_TOKEN	**KIE_SERVER_CONTROLLER_TOKEN**	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	–	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

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_MODE	**KIE_SERVER_MODE**	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**DEVELOPMENT**	False
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

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route for Decision Central. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	–	False
DECISION_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route for Decision Central. Leave blank for default hostname, e.g.: <application-name>-rhdmcentr-<project>.<default-domain-suffix>	–	False
KIE_SERVER_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route for KIE Server. 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 for KIE Server. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
DECISION_CENTRAL_HTTPS_SECRET	–	The name of the secret containing the keystore file for Decision Central.	decisioncentral-app-secret	True
DECISION_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret for Decision Central.	keystore.jks	False
DECISION_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate for Decision Central.	jboss	False
DECISION_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate for Decision Central.	mykeystorepass	False
KIE_SERVER_HTTPS_SECRET	–	The name of the secret containing the keystore file for KIE Server.	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret for KIE Server.	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate for KIE Server.	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate for KIE Server.	mykeystorepass	False
APPFORMER_JMS_BROKER_USER	**APPFORMER_JMS_BROKER_USER**	The user name to connect to the JMS broker.	jmsBrokerUser	True

Variable name	Image Environment Variable	Description	Example value	Required
APPFORMER_JMS_BROKER_PASSWORD	**APPFORMER_JMS_BROKER_PASSWORD**	The password to connect to the JMS broker.	–	True
DATAGRID_IMAGE	–	DataGrid image.	registry.redhat.io/jboss-datagrid-7/datagrid73-openshift:1.6	True
DATAGRID_CPU_LIMIT	–	DataGrid Container CPU limit.	1000m	True
DATAGRID_MEMORY_LIMIT	–	DataGrid Container memory limit.	2Gi	True
DATAGRID_VOLUME_CAPACITY	–	Size of the persistent storage for DataGrid's runtime data.	1Gi	True
AMQ_BROKER_IMAGE	–	AMQ Broker Image	registry.redhat.io/amq7/amq-broker:7.7	True
AMQ_ROLE	–	User role for standard broker user.	admin	True
AMQ_NAME	–	The name of the broker.	broker	True
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

Variable name	Image Environment Variable	Description	Example value	Required
AMQ_VOLUME_CAPACITY	–	Size of persistent storage for AMQ broker volume.	1Gi	True
AMQ_REPLICAS	–	Number of broker replicas for a cluster	2	True
DECISION_CENTRAL_CONTAINER_REPLICAS	–	Decision Central Container Replicas, defines how many Decision Central containers will be started.	2	True
KIE_SERVER_CONTAINER_REPLICAS	–	KIE Server Container Replicas, defines how many KIE Server containers will be started.	2	True
KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	**KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED**	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	false	False
KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	**KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE**	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	true	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	**KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL**	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	60000	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 need to modify this parameter only if you installed the ImageStreams in a different namespace/project.	openshift	True
DECISION_CENTRAL_IMAGE_STREAM_NAME	–	The name of the image stream to use for Decision Central. Default is "rhdm-decisioncentral-rhel8".	rhdm-decisioncentral-rhel8	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 "7.9.0".	7.9.0	True

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Maven mirror that Decision Central and 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:*,!repo-rhdmcentr	False
MAVEN_REPO_ID	**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	**MAVEN_REPO_URL**	Fully qualified URL to a Maven repository or service.	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_REPO_USERNAME	**MAVEN_REPO_USERNAME**	User name for accessing the Maven repository, if required.	–	False
MAVEN_REPO_PASSWORD	**MAVEN_REPO_PASSWORD**	Password to access the Maven repository, if required.	–	False
GIT_HOOKS_DIR	**GIT_HOOKS_DIR**	The directory to use for git hooks, if required.	**/opt/kie/data/git/hooks**	False
DECISION_CENTRAL_VOLUME_CAPACITY	–	Size of the persistent storage for Decision Central’s runtime data.	1Gi	True
DECISION_CENTRAL_MEMORY_LIMIT	–	Decision Central Container memory limit.	8Gi	True
DECISION_CENTRAL_JAVA_MAX_MEM_RATIO	**JAVA_MAX_MEM_RATIO**	Decision Central Container JVM max memory ratio. **-Xmx** is set to a ratio of the memory available on the container. The default is 80, which means the upper boundary is 80% of the available memory. To skip adding the **-Xmx** option, set this value to 0.	80	True
DECISION_CENTRAL_CPU_LIMIT	–	Decision Central Container CPU limit.	2000m	True
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	True

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CPU_LIMIT	–	KIE server Container CPU limit.	1000m	True
SSO_URL	**SSO_URL**	RH-SSO URL.	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name.	–	False
DECISION_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Decision Central RH-SSO Client name.	–	False
DECISION_CENTRAL_SSO_SECRET	**SSO_SECRET**	Decision Central RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	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 user name 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 user name.	preferred_username	False

Variable name	Image Environment Variable	Description	Example value	Required
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=example,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

Variable name	Image Environment Variable	Description	Example value	Required
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
AUTH_LDAP_PARSE_USERNAME	**AUTH_LDAP_PARSE_USERNAME**	A flag indicating if the DN is to be parsed for the user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. 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 user name. 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 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.	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 roleNameAttributeId 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

12.3.2. 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.

12.3.2.1. Services

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

Service	Port	Name	Description
${APPLICATION_NAME}-rhdmcentr	8080	http	All the Decision Central web server's ports.
	8443	https	
${APPLICATION_NAME}-rhdmcentr-ping	8888	ping	The JGroups ping port for rhdmcentr clustering.
${APPLICATION_NAME}-datagrid-ping	8888	ping	Provides a ping service for clustered applications.
${APPLICATION_NAME}-datagrid	11222	hotrod	Provides a service for accessing the application over Hot Rod protocol.
${APPLICATION_NAME}-kieserver	8080	http	All the KIE server web server's ports.
	8443	https	
${APPLICATION_NAME}-amq-tcp	61616	–	The broker's OpenWire port.
ping	8888	–	The JGroups ping port for amq clustering.

12.3.2.2. 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. See the Openshift documentation for more information.

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

12.3.2.3. 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. See the Openshift documentation for more information.

12.3.2.3.1. Triggers

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

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

12.3.2.3.2. 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. See the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-rhdmcentr	2
${APPLICATION_NAME}-kieserver	2

12.3.2.3.3. Pod Template

12.3.2.3.3.1. Service Accounts

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

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

12.3.2.3.3.2. Image

Deployment	Image
${APPLICATION_NAME}-rhdmcentr	**${DECISION_CENTRAL_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-kieserver	**${KIE_SERVER_IMAGE_STREAM_NAME}**

12.3.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/ready

${APPLICATION_NAME}-kieserver

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

12.3.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhdmcentr

> Http Get on http://localhost:8080/rest/healthy

${APPLICATION_NAME}-kieserver

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

12.3.2.3.3.5. Exposed Ports

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

12.3.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-rhdmcentr	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_ENABLED	–	true

Deployment	Variable name	Description	Example value
	KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED	If set to true, turns on KIE server global discovery feature (Sets the org.kie.server.controller.openshift.global.discovery.enabled system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_GLOBAL_DISCOVERY_ENABLED}**
	KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE	If OpenShift integration of Business Central is turned on, setting this parameter to true enables connection to KIE Server via an OpenShift internal Service endpoint. (Sets the org.kie.server.controller.openshift.prefer.kieserver.service system property)	**${KIE_SERVER_CONTROLLER_OPENSHIFT_PREFER_KIESERVER_SERVICE}**
	KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL	KIE ServerTemplate Cache TTL in milliseconds. (Sets the org.kie.server.controller.template.cache.ttl system property)	**${KIE_SERVER_CONTROLLER_TEMPLATE_CACHE_TTL}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	WORKBENCH_ROUTE_NAME	–	**${APPLICATION_NAME}-rhdmcentr**
	MAVEN_MIRROR_URL	Maven mirror that Decision Central and KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	**${MAVEN_MIRROR_URL}**

Deployment	Variable name	Description	Example value
	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.	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	User name for accessing the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	GIT_HOOKS_DIR	The directory to use for git hooks, if required.	**${GIT_HOOKS_DIR}**
	HTTPS_KEYSTORE_DIR	–	**/etc/decisioncentral-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret for Decision Central.	**${DECISION_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate for Decision Central.	**${DECISION_CENTRAL_HTTPS_NAME}**

Deployment	Variable name	Description	Example value
	HTTPS_PASSWORD	The password for the keystore and certificate for Decision Central.	**${DECISION_CENTRAL_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-rhdmcentr-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	APPFORMER_INFINISPAN_SERVICE_NAME	–	**${APPLICATION_NAME}-datagrid**
	APPFORMER_INFINISPAN_PORT	–	11222
	APPFORMER_JMS_BROKER_ADDRESS	–	**${APPLICATION_NAME}-amq-tcp**
	APPFORMER_JMS_BROKER_PORT	–	61616
	APPFORMER_JMS_BROKER_USER	The user name to connect to the JMS broker.	**${APPFORMER_JMS_BROKER_USER}**
	APPFORMER_JMS_BROKER_PASSWORD	The password to connect to the JMS broker.	**${APPFORMER_JMS_BROKER_PASSWORD}**
	JAVA_MAX_MEM_RATIO	Decision Central Container JVM max memory ratio. **-Xmx** is set to a ratio of the memory available on the container. The default is 80, which means the upper boundary is 80% of the available memory. To skip adding the **-Xmx** option, set this value to 0.	**${DECISION_CENTRAL_JAVA_MAX_MEM_RATIO}**
	SSO_URL	RH-SSO URL.	**${SSO_URL}**

Deployment	Variable name	Description	Example value
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**
	SSO_SECRET	Decision Central RH-SSO Client Secret.	**${DECISION_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Decision Central RH-SSO Client name.	**${DECISION_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name used to create the Client if it doesn't exist.	**${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 user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route for Decision Central. Leave blank for default hostname, e.g.: insecure-<application-name>-rhdmcentr-<project>.<default-domain-suffix>	**${DECISION_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route for Decision Central. Leave blank for default hostname, e.g.: <application-name>-rhdmcentr-<project>.<default-domain-suffix>	**${DECISION_CENTRAL_HOSTNAME_HTTPS}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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.	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. This option is used together with usernameBeginString and usernameEndString.	**${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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_END_STRING}**

Deployment	Variable name	Description	Example value
	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}**
	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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This parameter defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${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}**
${APPLICATION_NAME}-kieserver	**WORKBENCH_SERVICE_NAME**	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_ADMIN_USER	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret

Deployment	Variable name	Description	Example value
	KIE_SERVER_MODE	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**${KIE_SERVER_MODE}**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering. (Sets the org.drools.server.filter.classes system property)	**${DROOLS_SERVER_FILTER_CLASSES}**
	PROMETHEUS_SERVER_EXT_DISABLED	If set to false, the prometheus server extension will be enabled. (Sets the org.kie.prometheus.server.ext.disabled system property)	**${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)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws

Deployment	Variable name	Description	Example value
	KIE_SERVER_ID	–	–
	KIE_SERVER_ROUTE_NAME	–	insecure-${APPLICATION_NAME}-kieserver
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	MAVEN_MIRROR_URL	Maven mirror that Decision Central and KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for building and deploying your services.	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	Maven mirror configuration for KIE server.	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL
	RHDMCENTR_MAVEN_REPO_ID	–	repo-rhdmcentr
	RHDMCENTR_MAVEN_REPO_SERVICE	–	**${APPLICATION_NAME}-rhdmcentr**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	Set according to the credentials secret
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	Set according to the credentials secret

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.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	User name for accessing 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 for KIE Server.	**${KIE_SERVER_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate for KIE Server.	**${KIE_SERVER_HTTPS_NAME}**
	HTTPS_PASSWORD	The password for the keystore and certificate for KIE Server.	**${KIE_SERVER_HTTPS_PASSWORD}**
	KUBERNETES_NAMESPACE	–	–

Deployment	Variable name	Description	Example value
	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 user name 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.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route for KIE Server. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route for KIE Server. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER_HOSTNAME_HTTPS}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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.	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${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}**

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

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_ROLES_CTX_DN}**
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This parameter defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${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}**

12.3.2.3.3.7. Volumes

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

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

12.3.2.4. External Dependencies

12.3.2.4.1. Volume Claims

A **PersistentVolume** object is a storage resource in an OpenShift cluster. Storage is provisioned by an administrator by creating **PersistentVolume** objects from sources such as GCE Persistent Disks, AWS Elastic Block Stores (EBS), and NFS mounts. See the Openshift documentation for more information.

Name	Access Mode
${APPLICATION_NAME}-rhdmcentr-claim	ReadWriteMany

12.3.2.4.2. Secrets

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

decisioncentral-app-secret kieserver-app-secret

12.3.2.4.3. Clustering

Clustering in OpenShift EAP is achieved through one of two discovery mechanisms: Kubernetes or DNS. This is done by configuring the JGroups protocol stack in standalone-openshift.xml with either the **<openshift.KUBE_PING/>** or **<openshift.DNS_PING/>** elements. The templates are configured to use **DNS_PING**, however `KUBE_PING`is the default used by the image.

The discovery mechanism used is specified by the **JGROUPS_PING_PROTOCOL** environment variable which can be set to either **openshift.DNS_PING** or **openshift.KUBE_PING**. **openshift.KUBE_PING** is the default used by the image if no value is specified for **JGROUPS_PING_PROTOCOL**.

For DNS_PING to work, the following steps must be taken:

1. The **OPENSHIFT_DNS_PING_SERVICE_NAME** environment variable must be set to the name of the ping service for the cluster (see table above). If not set, the server will act as if it is a single-node cluster (a "cluster of one").

2. The **OPENSHIFT_DNS_PING_SERVICE_PORT** environment variables should be set to the port number on which the ping service is exposed (see table above). The **DNS_PING** protocol will attempt to discern the port from the SRV records, if it can, otherwise it will default to 8888.

3. A ping service which exposes the ping port must be defined. This service should be "headless" (ClusterIP=None) and must have the following:

    a. The port must be named for port discovery to work.

b. It must be annotated with **service.alpha.kubernetes.io/tolerate-unready-endpoints** set to **"true"**. Omitting this annotation will result in each node forming their own "cluster of one" during startup, then merging their cluster into the other nodes' clusters after startup (as the other nodes are not detected until after they have started).

Example ping service for use with DNS_PING

```
kind: Service
apiVersion: v1
spec:
    clusterIP: None
    ports:
    - name: ping
      port: 8888
    selector:
        deploymentConfig: eap-app
metadata:
    name: eap-app-ping
    annotations:
        service.alpha.kubernetes.io/tolerate-unready-endpoints: "true"
        description: "The JGroups ping port for clustering."
```

For **KUBE_PING** to work, the following steps must be taken:

1. The **OPENSHIFT_KUBE_PING_NAMESPACE** environment variable must be set (see table above). If not set, the server will act as if it is a single-node cluster (a "cluster of one").

2. The **OPENSHIFT_KUBE_PING_LABELS** environment variables should be set (see table above). If not set, pods outside of your application (albeit in your namespace) will try to join.

3. Authorization must be granted to the service account the pod is running under to be allowed to access Kubernetes' REST api. This is done on the command line.

Example 12.1. Policy commands

Using the default service account in the myproject namespace:

```
oc policy add-role-to-user view system:serviceaccount:myproject:default -n myproject
```

Using the eap-service-account in the myproject namespace:

```
oc policy add-role-to-user view system:serviceaccount:myproject:eap-service-account -n
myproject
```

12.4. RHDM79-KIESERVER.YAML TEMPLATE

Application template for a managed KIE Server, for Red Hat Decision Manager 7.9 - Deprecated

12.4.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	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 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 or service.	http://nexus.nexus-project.svc.cluster.local:8081/nexus/content/groups/public/	False

Variable name	Image Environment Variable	Description	Example value	Required
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	User name for accessing 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
CREDENTIALS_SECRET	–	Secret containing the KIE_ADMIN_USER and KIE_ADMIN_PWD values.	rhpam-credentials	True
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 need to modify this parameter only 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 "7.9.0".	7.9.0	True
KIE_SERVER_MODE	**KIE_SERVER_MODE**	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**PRODUCTION**	False
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

Variable name	Image Environment Variable	Description	Example value	Required
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
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

Variable name	Image Environment Variable	Description	Example value	Required
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_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	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-library=org.openshift.quickstarts:rhdm-kieserver-library:1.6.0-SNAPSHOT	False
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	False
SSO_URL	**SSO_URL**	RH-SSO URL.	https://rh-sso.example.com/auth	False
SSO_REALM	**SSO_REALM**	RH-SSO Realm name.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name 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 user name.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=example,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	LDAP Credentials used for authentication.	Password	False

Variable name	Image Environment Variable	Description	Example value	Required
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
AUTH_LDAP_SEARCH_TIME_LIMIT	**AUTH_LDAP_SEARCH_TIME_LIMIT**	The timeout in milliseconds for user or role searches.	10000	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. 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 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.	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 roleNameAttributeId 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 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

12.4.2. 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.

12.4.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. See 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.

12.4.2.2. 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. See 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}**

12.4.2.3. 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. See the Openshift documentation for more information.

12.4.2.3.1. Triggers

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

Deployment	Triggers
${APPLICATION_NAME}-kieserver	ImageChange

12.4.2.3.2. 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. See the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-kieserver	1

12.4.2.3.3. Pod Template

12.4.2.3.3.1. Service Accounts

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

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

12.4.2.3.3.2. Image

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

12.4.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

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

12.4.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

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

12.4.2.3.3.5. Exposed Ports

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

12.4.2.3.3.6. 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	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_SERVER_MODE	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**${KIE_SERVER_MODE}**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${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_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	MAVEN_MIRROR_URL	Maven mirror that KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for deploying your services.	**${MAVEN_MIRROR_URL}**
	MAVEN_MIRROR_OF	Maven mirror configuration for KIE server.	**${MAVEN_MIRROR_OF}**
	MAVEN_REPOS	–	RHDMCENTR,EXTERNAL

Deployment	Variable name	Description	Example value
	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.	**${DECISION_CENTRAL_SERVICE}**
	RHDMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHDMCENTR_MAVEN_REPO_USERNAME	–	Set according to the credentials secret
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	Set according to the credentials secret
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository. If set, it can be excluded from the optionally configured mirror by adding it to MAVEN_MIRROR_OF. For example: external:*,!repo-rhdmcentr,!repo-custom. If MAVEN_MIRROR_URL is set but MAVEN_MIRROR_ID is not set, an id will be generated randomly, but won't be usable in MAVEN_MIRROR_OF.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	User name for accessing the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**

Deployment	Variable name	Description	Example value
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_MGMT_DISABLED	Disable management api and don't allow KIE containers to be deployed/undeployed or started/stopped. Sets the property org.kie.server.mgmt.api.disabled to true and org.kie.server.startup.strategy to LocalContainersStartupStrategy.	**${KIE_SERVER_MGMT_DISABLED}**
	KIE_SERVER_STARTUP_STRATEGY	–	OpenShiftStartupStrategy
	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.	**${KIE_SERVER_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-kieserver-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL.	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war

Deployment	Variable name	Description	Example value
	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.	**${KIE_SERVER_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name used to create the Client if it doesn't exist.	**${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 user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: insecure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${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>	**${KIE_SERVER_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**

Deployment	Variable name	Description	Example value
	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}**
	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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. This option is used together with usernameBeginString and usernameEndString.	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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 user name. 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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_ROLES_CTX_DN}**
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_ROLE_NAME_ATTRIBUTE_ID}**
	AUTH_LDAP_PARSE_ROLE_NAME_FROM_DN	A flag indicating if the DN returned by a query contains the roleNameAttributeID. If set to true, the DN is checked for the roleNameAttributeID. If set to false, the DN is not checked for the roleNameAttributeID. This flag can improve the performance of LDAP queries.	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${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}**

12.4.2.3.3.7. Volumes

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

12.4.2.4. External Dependencies

12.4.2.4.1. Secrets

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

kieserver-app-secret

12.5. RHDM79-PROD-IMMUTABLE-KIESERVER.YAML TEMPLATE

Application template for an immutable KIE server in a production environment, for Red Hat Decision Manager 7.9 - Deprecated

12.5.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
CREDENTIALS_SECRET	–	Secret containing the KIE_ADMIN_USER and KIE_ADMIN_PWD values.	rhpam-credentials	True
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 need to modify this parameter only 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 "7.9.0".	7.9.0	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:1.6.0-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.	master	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_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

Variable name	Image Environment Variable	Description	Example value	Required
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
MAVEN_REPO_USERNAME	**EXTERNAL_MAVEN_REPO_USERNAME**	User name for accessing 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

Variable name	Image Environment Variable	Description	Example value	Required
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
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 user name used to create the Client if it doesn’t exist.	–	False

Variable name	Image Environment Variable	Description	Example value	Required
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 user name.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication.	uid=admin,ou=users,ou=example,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

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. 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 user name. 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 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.	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 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

12.5.2. 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.

12.5.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. See 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.

12.5.2.2. 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. See 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}**

12.5.2.3. 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-rhel8:7.9.0	**rhpam-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub, Generic, ImageChange, ConfigChange

12.5.2.4. 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. See the Openshift documentation for more information.

12.5.2.4.1. Triggers

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

Deployment	Triggers
${APPLICATION_NAME}-kieserver	ImageChange

12.5.2.4.2. 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. See the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-kieserver	2

12.5.2.4.3. Pod Template

12.5.2.4.3.1. Service Accounts

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

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

12.5.2.4.3.2. Image

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

12.5.2.4.3.3. Readiness Probe

${APPLICATION_NAME}-kieserver

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

12.5.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

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

12.5.2.4.3.5. Exposed Ports

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

12.5.2.4.3.6. 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	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	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}**
	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	–	–

Deployment	Variable name	Description	Example value
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	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	–	Set according to the credentials secret
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	Set according to the credentials secret

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	User name for accessing 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 user name 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.	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${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. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**

12.5.2.4.3.7. Volumes

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

12.5.2.5. External Dependencies

12.5.2.5.1. Secrets

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

kieserver-app-secret

12.6. RHDM79-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 7.9 - Deprecated

12.6.1. Parameters

Templates allow you to define parameters that 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. See the Openshift documentation for more information.

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	–	The name for the application.	myapp	True
CREDENTIALS_SECRET	–	Secret containing the KIE_ADMIN_USER and KIE_ADMIN_PWD values.	rhpam-credentials	True
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 need to modify this parameter only 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 "7.9.0".	7.9.0	True

Variable name	Image Environment Variable	Description	Example value	Required
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
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

Variable name	Image Environment Variable	Description	Example value	Required
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:1.6.0-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	master	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 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**	User name for accessing 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

Variable name	Image Environment Variable	Description	Example value	Required
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
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

Variable name	Image Environment Variable	Description	Example value	Required
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
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

Variable name	Image Environment Variable	Description	Example value	Required
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
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:7.7	True

Variable name	Image Environment Variable	Description	Example value	Required
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 need to modify this parameter only 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
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 user name 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 user name.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP endpoint to connect for authentication. For failover, set two or more LDAP endpoints separated by space.	ldap://myldap.example.com:389	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication	uid=admin,ou=users,ou=example,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

Variable name	Image Environment Variable	Description	Example value	Required
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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. 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 user name. 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 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.	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

12.6.2. 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.

12.6.2.1. Services

A service is an abstraction which defines a logical set of pods and a policy by which to access them. See 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.
${APPLICATION_NAME}-amq-jolokia	8161	amq-jolokia-console	The broker's console and Jolokia port.

Service	Port	Name	Description
${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.

12.6.2.2. 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. See 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>
${APPLICATION_NAME}-amq-tcp-ssl	TLS passthrough	<default>

12.6.2.3. 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-rhel8:7.9.0	**rhpam-7/rhdm-kieserver-rhel8**	**${APPLICATION_NAME}-kieserver:latest**	GitHub, Generic, ImageChange, ConfigChange

12.6.2.4. 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. See the Openshift documentation for more information.

12.6.2.4.1. Triggers

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

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

12.6.2.4.2. 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. See the container-engine documentation for more information.

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

12.6.2.4.3. Pod Template

12.6.2.4.3.1. Service Accounts

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

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

12.6.2.4.3.2. Image

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

12.6.2.4.3.3. 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

12.6.2.4.3.4. Liveness Probe

${APPLICATION_NAME}-kieserver

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

12.6.2.4.3.5. Exposed Ports

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-kieserver	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-amq	console-jolokia	8161	**TCP**
	amq-amqp	5672	**TCP**
	amqp-ssl	5671	**TCP**

Deployments	Name	Port	Protocol
	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**

12.6.2.4.3.6. 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	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	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	–	**${APPLICATION_NAME}-kieserver**
	KIE_SERVER_CONTAINER_DEPLOYMENT	KIE Server Container deployment configuration with optional alias. Format: containerId=groupId:artifactId:version|c2(alias2)=g2:a2:v2	**${KIE_SERVER_CONTAINER_DEPLOYMENT}**
	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	–	Set according to the credentials secret

Deployment	Variable name	Description	Example value
	RHDMCENTR_MAVEN_REPO_PASSWORD	–	Set according to the credentials secret
	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	User name for accessing 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	**${KIE_SERVER_JMS_QUEUE_REQUEST}**
	KIE_SERVER_JMS_QUEUE_RESPONSE	JNDI name of response queue for JMS. The default value is queue/KIE.SERVER.RESPONSE	**${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}**

Deployment	Variable name	Description	Example value
	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.	**${AMQ_QUEUES}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret	**${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 user name 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	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${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. For failover, set two or more LDAP endpoints separated by space.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication	**${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 user name. If set to true, the DN is parsed for the user name. If set to false the DN is not parsed for the user name. 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 user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the user name. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${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}**

12.6.2.4.3.7. 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

12.6.2.5. External Dependencies

12.6.2.5.1. Secrets

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

kieserver-app-secret broker-app-secret

12.7. 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.

PART III. IMPLEMENTING HIGH AVAILABLE EVENT-DRIVEN DECISIONING USING THE DECISION ENGINE ON RED HAT OPENSHIFT CONTAINER PLATFORM

As a business rules developer, you can use high available event-driven decisioning, including Complex Event Processing (CEP), in your code that uses the decision engine. You can implement high available event-driven decisioning on Red Hat OpenShift Container Platform.

You cannot use a standard deployment of Red Hat Decision Manager on Red Hat OpenShift Container Platform, as described in *Deploying a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform using Operators*, to implement high available event-driven decisioning, because the standard deployment supports only stateless processing. You must therefore create a custom implementation using the provided reference implementation.

Prerequisites

- A Red Hat OpenShift Container Platform version 4 environment is available. For the exact versions of Red Hat OpenShift Container Platform that the current release supports, see Red Hat Process Automation Manager 7 Supported Configurations.
- A Kafka Cluster is deployed in the OpenShift environment with Red Hat AMQ Streams.
- The OpenJDK Java development environment is installed.
- Maven, Docker, and kubectl are installed.
- The **oc** OpenShift command line tool is installed.

CHAPTER 13. HIGH AVAILABLE EVENT-DRIVEN DECISIONING ON RED HAT OPENSHIFT CONTAINER PLATFORM

Use the decision engine to implement high available event-driven decisioning on Red Hat OpenShift Container Platform.

An *event* models a fact that happened in a specific point in time. The decision engine offers a rich set of temporal operators to compare, correlate, and accumulate events. In event-driven decisioning, the decision engine processes complex series of decisions based on events. Every event can alter the state of the engine, influencing decisions for subsequent events.

You cannot use a standard deployment of Red Hat Decision Manager on Red Hat OpenShift Container Platform, as described in Part I, “Deploying a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform using Operators”, to run high available event-driven decisioning. The deployment includes KIE Server pods, which remain independent of each other when scaled. The states of the pods are not synchronized. Therefore, only stateless calls can be processed reliably.

The Complex Event Processing (CEP) API is useful for event-driven decisioning with the decision engine. The decision engine uses CEP to detect and process multiple events within a collection of events, to uncover relationships that exist between events, and to infer new data from the events and their relationships. For more information about CEP in the decision engine, see *Decision engine in Red Hat Decision Manager*.

Implement high available event-driven decisioning on Red Hat OpenShift Container Platform based on the reference implementation provided with Red Hat Decision Manager. This implementation provides an environment with safe failover.

In this reference implementation, you can scale the pod with the processing code. The replicas of the pod are not independent. One of the replicas is automatically designated *leader*. If the leader ceases to function, another replica is automatically made leader and the processing continues without interruption or data loss.

The election of the leader is implemented with Kubernetes ConfigMaps. Coordination of the leader with other replicas is performed with exchanged messages through Kafka. The leader is always the first to process an event. When processing is complete, the leader notifies other replicas. A replica that is not the leader starts executing an event only after it has been completely processed on the leader.

When a new replica joins the cluster, this replica requests a snapshot of the current Drools session from the leader. The leader can use a recent existing snapshot if one is available in a Kafka topic. If a recent snapshot is not available, the leader produces a new snapshot on demand. After receiving the snapshot, the new replica deserializes it and eventually executes the last events not included in the snapshot before starting to process new events in coordination with the leader.

With the default implementation method, the service is built into the HA CEP server as a fat KJAR. In this case, build and deploy the server again to change the version of the service. The content of the working memory is lost when you switch to the new version. For instructions about the default implementation method, see Chapter 14, *Implementing the HA CEP server* .

If you require upgrading versions of the service without losing the content of the working memory, use an alternate implementation method and provide the KJAR and all dependencies in a Maven repository. In this implementation method, use an **UpdateKJarGAV** call from the client code to trigger deployment of a new KJAR version. This call is processed by the leader and then other replicas, and each of the pods then loads the new KJAR. The contents of the working memory remain in place. For instructions about this implementation method, see Chapter 15, *Implementing the HA CEP server with a Maven repository for updating the KJAR service*.

CHAPTER 14. IMPLEMENTING THE HA CEP SERVER

The high-availability (HA) CEP server runs on the Red Hat OpenShift Container Platform environment. It includes all necessary Drools rules and other code required to process events.

Prepare the source, build it, and then deploy it on Red Hat OpenShift Container Platform.

Alternatively, use a different process to deploy the HA CEP server where you can update the KJAR service at any time. For instructions about this process, see Chapter 15, *Implementing the HA CEP server with a Maven repository for updating the KJAR service*.

Prerequisites

- You are logged into the project with administrator privilege using the **oc** command-line tool.

Procedure

1. Download the **rhdm-7.9.1-reference-implementation.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

2. Extract the contents of the file and then uncompress the **rhdm-7.9.1-openshift-drools-hacep-distribution.zip** file.

3. Change to the **openshift-drools-hacep-distribution/sources** directory.

4. Review and modify the server code based on the sample project in the **sample-hacep-project/sample-hacep-project-kjar** directory. The complex event processing logic is defined by the DRL rules in the **src/main/resources/org/drools/cep** subdirectory.

5. Build the project using the standard Maven command:

```
mvn clean install -DskipTests
```

6. Enable the OpenShift operator for Red Hat AMQ Streams and then create an AMQ Streams (kafka) cluster in the project. For information about installing Red Hat AMQ Streams, see *Using AMQ Streams on OpenShift*.

7. To create the kafka topics that are required for operation of the server, remain in the **openshift-drools-hacep-distribution/sources** directory and run the following commands:

```
oc apply -f kafka-topics/control.yaml
oc apply -f kafka-topics/events.yaml
oc apply -f kafka-topics/kiesessioninfos.yaml
oc apply -f kafka-topics/snapshot.yaml
```

8. In order to enable application access to the ConfigMap that is used in the leader election, configure role-based access control. Change to the **springboot** directory and enter the following commands:

```
oc create -f kubernetes/service-account.yaml
oc create -f kubernetes/role.yaml
oc create -f kubernetes/role-binding.yaml
```

For more information about configuring role-based access control in Red Hat OpenShift Container Platform, see Using RBAC to define and apply permissions in the Red Hat OpenShift Container Platform product documentation.

9. In the **springboot** directory, enter the following commands to create the image for deployment and push it into the repository configured for your OpenShift environment:

   ```
   oc new-build --binary --strategy=docker --name openshift-kie-springboot
   oc start-build openshift-kie-springboot --from-dir=. --follow
   ```

10. Enter the following command to detect the name of the image that was built:

    ```
    oc get is/openshift-kie-springboot -o template --template='{{range .status.tags}}{{range
    .items}}{{.dockerImageReference}}{{end}}{{end}}'
    ```

11. Open the **kubernetes/deployment.yaml** file in a text editor.

12. Replace the existing image URL with the result of the previous command.

13. Remove all characters at the end of the line starting with the @ symbol, then add **:latest** to the line. For example:

    ```
    image: image-registry.openshift-image-registry.svc:5000/hacep/openshift-kie-
    springboot:latest
    ```

14. Save the file.

15. Enter the following command to deploy the image:

    ```
    oc apply -f kubernetes/deployment.yaml
    ```

CHAPTER 15. IMPLEMENTING THE HA CEP SERVER WITH A MAVEN REPOSITORY FOR UPDATING THE KJAR SERVICE

You can implement the HA CEP server that retrieves the KJAR service and all dependencies from a Maven repository that you provide. In this case, you can update the KJAR service at any time by updating it in the Maven repository and then making a call from the client code.

Prepare the source, build it, and then deploy it on Red Hat OpenShift Container Platform. Set certain environment variables in the **deployment.yaml** file before deploying the server. To use a Maven repository, you must set the **UPDATABLEKJAR** variable to **true**.

Prerequisites

- You are logged into the project with administrator privilege using the **oc** command-line tool.
- You configured a Maven repository that is accessible from your Red Hat OpenShift Container Platform environment.

Procedure

1. Download the **rhdm-7.9.1-reference-implementation.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
2. Extract the contents of the file and then uncompress the **rhdm-7.9.1-openshift-drools-hacep-distribution.zip** file.
3. Change to the **openshift-drools-hacep-distribution/sources** directory.
4. Review and modify the server code based on the sample project in the **sample-hacep-project/sample-hacep-project-kjar** directory. The complex event processing logic is defined by the DRL rules in the **src/main/resources/org/drools/cep** subdirectory.
5. Build the project using the standard Maven command:

   ```
   mvn clean install -DskipTests
   ```

   Upload the resulting KJAR and any required dependencies to the Maven repository.
6. Enable the OpenShift operator for Red Hat AMQ Streams and then create an AMQ Streams (kafka) cluster in the project. For information about installing Red Hat AMQ Streams, see *Using AMQ Streams on OpenShift*.
7. To create the kafka topics that are required for operation of the server, remain in the **openshift-drools-hacep-distribution/sources** directory and run the following commands:

   ```
   oc apply -f kafka-topics/control.yaml
   oc apply -f kafka-topics/events.yaml
   oc apply -f kafka-topics/kiesessioninfos.yaml
   oc apply -f kafka-topics/snapshot.yaml
   ```
8. In order to enable application access to the ConfigMap that is used in the leader election, configure role-based access control. Change to the **springboot** directory and enter the following commands:

```
oc create -f kubernetes/service-account.yaml
oc create -f kubernetes/role.yaml
oc create -f kubernetes/role-binding.yaml
```

For more information about configuring role-based access control in Red Hat OpenShift Container Platform, see Using RBAC to define and apply permissions in the Red Hat OpenShift Container Platform product documentation.

9. In the **springboot** directory, edit the **pom.xml** file to remove the following dependency:

```
<dependency>
    <groupId>org.kie</groupId>
    <artifactId>sample-hacep-project-kjar</artifactId>
</dependency>
```

10. In the **springboot** directory, enter the following commands to create the image for deployment and push it into the repository configured for your OpenShift environment:

```
oc new-build --binary --strategy=docker --name openshift-kie-springboot
oc start-build openshift-kie-springboot --from-dir=. --follow
```

11. Enter the following command to detect the name of the image that was built:

```
oc get is/openshift-kie-springboot -o template --template='{{range .status.tags}}{{range
.items}}{{.dockerImageReference}}{{end}}{{end}}'
```

12. Open the **kubernetes/deployment.yaml** file in a text editor.

13. Replace the existing image URL with the result of the previous command.

14. Remove all characters at the end of the line starting with the **@** symbol, then add **:latest** to the line. For example:

```
image: image-registry.openshift-image-registry.svc:5000/hacep/openshift-kie-
springboot:latest
```

15. Under the **containers:** line and the **env:** line, set environment variables as in the following example:

```
containers:
  - env:
    - name: UPDATABLEKJAR
      value: "true"
    - name: KJARGAV
      value: <GroupID>:<ArtifactID>:<Version>
    - name: MAVEN_LOCAL_REPO
      value: /app/.m2/repository
    - name: MAVEN_MIRROR_URL
      value: http://<nexus_url>/repository/maven-releases/
    - name: MAVEN_SETTINGS_XML
      value: /app/.m2/settings.xml
```

In this example, replace the value of the **KJARGAV** variable with the group, artifact, and version (GAV) of your KJAR service and the value of the **MAVEN_MIRROR_URL** variable with the URL to the Maven repository that contains your KJAR service.

Optionally, set other variables. For a list of supported environment variables, see Section 15.1, "Optional environment variables supported by the HA CEP server".

16. Save the file.

17. Enter the following command to deploy the image:

```
oc apply -f kubernetes/deployment.yaml
```

For instructions about triggering a KJAR update from the client code, see Chapter 16, *Creating the HA CEP client*.

15.1. OPTIONAL ENVIRONMENT VARIABLES SUPPORTED BY THE HA CEP SERVER

The following table lists optional environment variables that you can set for an HA CEP server that is configured to use a Maven repository. Add these variables to the **deployment.yaml** file to set them at deployment time.

NOTE

To use a Maven repository, ensure that you set the **UPDATABLEKJAR** and **KJARGAV** environment variables for the server, as described in Chapter 15, *Implementing the HA CEP server with a Maven repository for updating the KJAR service*.

Table 15.1. Optional environment variables supported by the HA CEP server

Name	Description	Example
MAVEN_LOCAL_REPO	Directory to use as the local Maven repository.	**/root/.m2/repository**
MAVEN_MIRROR_URL	The base URL of a Maven mirror that can be used for retrieving artifacts.	**http://nexus3-my-kafka-project.192.168.99.133.nip.io/repository/maven-public/**
MAVEN_MIRRORS	If set, multi-mirror support is enabled. The value contains a list of mirror prefixes, divided by commas. If this variable is set, the names of other **MAVEN_MIRROR_*** variables must contain a prefix, for example, **DEV_MAVEN_MIRROR_URL** and **QE_MAVEN_MIRROR_URL**	**DEV,QE**

Name	Description	Example
MAVEN_REPOS	If set, multi-repo support is enabled. The value contains a list of repo prefixes, divided by commas. If this variable is set, the names of other **MAVEN_REPO_*** variables must contain a prefix, for example, **DEV_MAVEN_REPO_URL** and **QE_MAVEN_REPO_URL**.	**DEV,QE**
MAVEN_SETTINGS_XML	The location of a custom Maven **settings.xml** file to use	**/root/.m2/settings.xml**
***prefix*_MAVEN_MIRROR_ID**	Identifier to be used for the specified mirror. If omitted, a unique ID is generated.	**internal-mirror**
***prefix*_MAVEN_MIRROR_OF**	Repository IDs mirrored by this mirror. Defaults to **external:***	**external:*,!my-repo**
***prefix*_MAVEN_MIRROR_URL**	The URL of the mirror	**http://10.0.0.1:8080/repository/internal**
***prefix*_MAVEN_REPO_HOST**	Maven repository host name	**repo.example.com**
***prefix*_MAVEN_REPO_ID**	Maven repository ID	**my-repo**
***prefix*_MAVEN_REPO_LAYOUT**	Maven repository layout	**default**
***prefix*_MAVEN_REPO_USERNAME**	Maven repository username	**mavenUser**
***prefix*_MAVEN_REPO_PASSPHRASE**	Maven repository passphrase	**maven1!**
***prefix*_MAVEN_REPO_PASSWORD**	Maven repository password	**maven1!**
***prefix*_MAVEN_REPO_PATH**	Maven repository path	**/maven2/**
***prefix*_MAVEN_REPO_PORT**	Maven repository port	**8080**
***prefix*_MAVEN_REPO_PRIVATE_KEY**	Local path to a private key for connecting to the Maven repository	**${user.home}/.ssh/id_dsa**

Name	Description	Example
***prefix*_MAVEN_REPO_PROTOCOL**	Maven repository protocol	**http**
***prefix*_MAVEN_REPO_RELEASES_ENABLED**	Maven repository releases enabled	**true**
***prefix*_MAVEN_REPO_RELEASES_UPDATE_POLICY**	Maven repository releases update policy	**always**
***prefix*_MAVEN_REPO_SERVICE**	Maven repository OpenShift service. This value is used if a URL or host/port/protocol is not specified.	**buscentr-myapp**
***prefix*_MAVEN_REPO_SNAPSHOTS_ENABLED**	Maven repository snapshots enabled	**true**
***prefix*_MAVEN_REPO_SNAPSHOTS_UPDATE_POLICY**	Maven repository snapshots update policy	**always**
***prefix*_MAVEN_REPO_URL**	Fully qualified URL for the Maven repository	**http://repo.example.com:8080/maven2/**

CHAPTER 16. CREATING THE HA CEP CLIENT

You must adapt your CEP client code to communicate with the HA CEP server image. Use the sample project included in the reference implementation for your client code. You can run your client code inside or outside the OpenShift environment.

Procedure

1. Download the **rhdm-7.9.1-reference-implementation.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

2. Extract the contents of the file and then uncompress the **rhdm-7.9.1-openshift-drools-hacep-distribution.zip** file.

3. Change to the **openshift-drools-hacep-distribution/sources** directory.

4. Review and modify the client code based on the sample project in the **sample-hacep-project/sample-hacep-project-client** directory. Ensure that the code fulfills the additional requirements described in Chapter 17, *Requirements for HA CEP client and server code* .

5. To update the KJAR version in an implementation that uses the method described in Chapter 15, *Implementing the HA CEP server with a Maven repository for updating the KJAR service*, add an **UpdateKJarGAV** call to the client, similar to the following code:

```
        TopicsConfig envConfig = TopicsConfig.getDefaultTopicsConfig();
        Properties props = getProperties();
        try (RemoteStreamingKieSession producer =
RemoteStreamingKieSession.create(props, envConfig)){
            producer.updateKJarGAV("org.kie:fake-jar:0.1");
        }
```

   Ensure that a KJAR with the specified GAV is available in the Maven repository when this call is executed.

6. In the **sample-hacep-project/sample-hacep-project-client** directory, generate a keystore, using **password** as a password. Enter the following command:

```
keytool -genkeypair -keyalg RSA -keystore src/main/resources/keystore.jks
```

7. Extract the HTTPS certificate from the OpenShift environment and add it to the keystore. Enter the following commands:

```
oc extract secret/my-cluster-cluster-ca-cert --keys=ca.crt --to=- > src/main/resources/ca.crt
keytool -import -trustcacerts -alias root -file src/main/resources/ca.crt -keystore
src/main/resources/keystore.jks -storepass password -noprompt
```

8. In the **src/main/resources** subdirectory of the project, open the **configuration.properties** file and replace **<bootstrap-hostname>** with the address that the route for the Kafka server provides.

9. Build the project using the standard Maven command:

```
mvn clean install
```

10. Change the **sample-hacep-project-client** project directory and enter the following command to run the client:

    ```
    mvn exec:java -Dexec.mainClass="org.kie.hacep.sample.client.ClientProducerDemo"
    ```

    This command executes the **main** method of the **ClientProducerDemo** class.

CHAPTER 17. REQUIREMENTS FOR HA CEP CLIENT AND SERVER CODE

When developing client and server code for high-availability CEP, follow certain additional requirements.

kie-remote API

The client code must use the **kie-remote** API and not the **kie** API. The **kie-remote** API is specified in the **org.kie:kie-remote** Maven artifact. You can find the source code in the **kie-remote** Maven module.

Explicit timestamps

The decision engine needs to determine the sequence in which events happen. For this reason, every event must have an associated timestamp assigned to it. In a high-availability environment, make this timestamp a property of the JavaBean that models the event. Annotate the event class with the **@Timestamp** annotation, where the name of the timestamp attribute itself is the parameter, as in the following example:

```
@Role(Role.Type.EVENT)
@Timestamp("myTime")
public class StockTickEvent implements Serializable {

    private String company;
    private double price;
    private long myTime;
}
```

If you do not provide a timestamp attribute, Drools assigns a timestamp to every event based on the time when the event is inserted by the client into a remote session. However, this mechanism depends on the clocks on the client machines. If clocks between different clients diverge, inconsistencies can occur between events inserted by these hosts.

Lambda expressions for non-memory actions

Working memory actions (actions to insert, modify, or delete information in the working memory of the decision engine) must be processed on every node of the cluster. Actions that are not memory actions must be executed only on the leader.

For example, the code might include the following rule:

```
rule FindAdult when
  $p : Person(age >= 18)
then
  modify($p) { setAdult(true) }; // working memory action
  sendEmailTo($p); // side effect
end
```

When this rule is triggered, the person must be marked as an adult on every node. However, only the leader must send the email, so that only one copy of the email is sent.

Therefore, in your code, wrap the email action (called a *side effect*) in a lambda expression, as shown in the following example:

```
rule FindAdult when
  $p : Person(age >= 18)
then
```

```
  modify($p) { setAdult(true) };
  DroolsExecutor.getInstance().execute( () -> sendEmailTo($p) );
end
```

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Tuesday, March 8, 2022.

APPENDIX B. CONTACT INFORMATION

Red Hat Decision Manager documentation team: brms-docs@redhat.com