



Red Hat Decision Manager 7.1

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

Red Hat Decision Manager 7.1 Deploying a Red Hat Decision Manager immutable server environment on Red Hat OpenShift Container Platform

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

Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to deploy a Red Hat Decision Manager 7.1 immutable server environment on Red Hat OpenShift Container Platform.

Table of Contents

PREFACE	3
CHAPTER 1. OVERVIEW OF RED HAT DECISION MANAGER ON RED HAT OPENSIFT CONTAINER PLATFORM	4
CHAPTER 2. PREPARING TO DEPLOY RED HAT DECISION MANAGER IN YOUR OPENSIFT ENVIRONMENT	6
2.1. ENSURING THE AVAILABILITY OF IMAGE STREAMS	6
2.2. CREATING THE SECRETS FOR DECISION SERVER	6
CHAPTER 3. ENVIRONMENT WITH IMMUTABLE SERVERS	8
3.1. EXTRACTING THE SOURCE CODE FROM DECISION CENTRAL FOR USE IN AN S2I BUILD	8
3.2. DEPLOYING AN IMMUTABLE DECISION SERVER	8
APPENDIX A. VERSIONING INFORMATION	12

PREFACE

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

Prerequisites

- At least two gigabytes of memory must be available in the OpenShift cluster/namespace.
- The OpenShift project for the deployment must be created.
- You must be logged in to the project using the **oc** command. For more information about the **oc** command-line tool, see the OpenShift [CLI Reference](#). If you want to use the OpenShift Web console to deploy templates, you must also be logged on using the Web console.

CHAPTER 1. OVERVIEW OF RED HAT DECISION MANAGER ON RED HAT OPENSIFT 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, providing as few or as many containers as necessary for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.

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

- Decision Server, also known as *Execution Server* or *KIE Server*, is the infrastructure element that runs decision services and other deployable assets (collectively referred to as *services*). All logic of the services runs on execution servers.

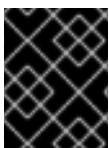
You can freely scale up a Decision Server pod, providing as many copies as necessary, running on the same host or different hosts. As you scale a pod up or down, all its copies run the same services. OpenShift provides load balancing and a request can be handled by any of the pods.

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

- Decision Central is a web-based interactive environment for authoring services. It also provides a management console. You can use Decision Central to develop services and deploy them to Decision Servers.

Decision Central is a centralized application. However, you can configure it for high availability, where multiple pods run and share the same data.

Decision 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, Decision Central can place the compiled services (KJAR files) into the built-in Maven repository or (if configured) into an external Maven repository.



IMPORTANT

In the current version, high-availability Decision Central functionality is a technology preview.

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

The following environment types are typical:

- *Authoring or managed environment*: An environment architecture that can be used for creating and modifying services using Decision Central and also for running services on Decision Servers. It consists of pods that provide Decision Central for the authoring work and one or more Decision Servers for execution of the services. Each Decision Server is a pod that you can replicate by scaling it up or down as necessary. You can deploy and undeploy services on each Decision Server using Decision Central. For instructions about deploying this environment, see [Deploying a Red Hat Decision Manager authoring or managed server environment on Red Hat OpenShift Container Platform](#).
- *Deployment with immutable servers*: An alternate environment for running existing services for staging and production purposes. In this environment, when you deploy a Decision Server pod, it builds an image that loads and starts a service or group of services. You cannot stop any service

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

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

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

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

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

2.1. ENSURING THE AVAILABILITY OF IMAGE STREAMS

You must ensure that the image streams that are required for the deployment are available in your OpenShift environment. Some versions of the OpenShift environment include the necessary image streams. You must check if they are available. If they are not available, you must install the **rhdm71-image-streams.yaml** file.

Procedure

1. Run the following commands:

```
$ oc get imagestreamtag -n openshift | grep rhdm71-decisioncentral-openshift
$ oc get imagestreamtag -n openshift | grep rhdm71-kieserver-openshift
```

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

2. If the output of one or both of the commands is empty, download the **rhdm-7.1.0-openshift-templates.zip** product deliverable file from the [Software Downloads](#) page. Extract the **rhdm71-image-streams.yaml** file from it. Complete one of the following actions:

- Run the following command:

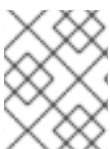
```
$ oc create -f rhdm71-image-streams.yaml
```

- Using the OpenShift Web UI, select **Add to Project** → **Import YAML / JSON**, then choose the file or paste its contents.

2.2. CREATING THE SECRETS FOR DECISION SERVER

OpenShift uses objects called **Secrets** to hold sensitive information, such as passwords or keystores. See the [Secrets chapter](#) in the OpenShift documentation for more information.

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



NOTE

You do not need to create the secrets object if you are planning to deploy only Decision Servers without support for HTTPS.

Procedure

1. Generate an SSL keystore with a private and public key for SSL encryption for Decision Server.

In a production environment, generate a valid signed certificate that matches the expected URL of the Decision Server. Save the keystore in a file named **keystore.jks**. Record the name of the certificate and the password of the keystore file.

See [Generate a SSL Encryption Key and Certificate](#) for more information on how to create a keystore with self-signed or purchased SSL certificates.

2. 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
```

CHAPTER 3. ENVIRONMENT WITH IMMUTABLE SERVERS

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

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

When you create a Decision Server image, you must 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 Decision Central, copy the source into a separate repository for the S2I build. OpenShift automatically builds the source, installs the process into the Decision Server image, and starts the process. No further management of the image is required. If you want to use a new version of the process, you can build a new image.

If you are using Decision 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.

3.1. EXTRACTING THE SOURCE CODE FROM DECISION CENTRAL FOR USE IN AN S2I BUILD

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

Procedure

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

```
git clone ssh://adminUser@decision-central-  
host:8001/MySpace/MyProject
```

Replace:

- **adminUser** with the administrative user for Decision Central
- **decision-central-host** with the host on which Decision Central is running
- **MySpace** with the name of the Decision Central space in which the project is located
- **MyProject** with the name of the project

2. Upload the source code to another Git repository for the S2I build.

3.2. DEPLOYING AN IMMUTABLE DECISION SERVER

To deploy an immutable Decision Server, use the **rhdm71-prod-immutable-kieserver.yaml** template file. You can extract this file from the **rhdm-7.1.0-openshift-templates.zip** product deliverable file. You can download the file from the [Software Downloads](#) page.

When you deploy an immutable Decision 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.

Procedure

1. 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 **rhdm71-prod-immutable-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>/rhdm71-prod-immutable-
kieserver.yaml -p KIE_SERVER_HTTPS_SECRET=kieserver-app-secret
```

In this command line:

- 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. You can view the template file to see descriptions for all parameters.

2. Set the following parameters as necessary:

- **KIE Server Keystore Secret Name (KIE_SERVER_HTTPS_SECRET)**: The name of the secret for Decision Server, as created in [Section 2.2, “Creating the secrets for Decision Server”](#).
- **Application Name (APPLICATION_NAME)**: The name of the OpenShift application. It is used in the default URL for Decision Server. OpenShift uses the application name to create a separate set of deployment configurations, services, routes, labels, and artifacts. You can deploy several applications using the same template into the same project, as long as you use different application names. Also, the application name determines the name of the server configuration (server template) on the Decision Central that the Decision Server is to join.
- **KIE Server Certificate Name (KIE_SERVER_HTTPS_NAME)**: The name of the certificate in the keystore that you created in [Section 2.2, “Creating the secrets for Decision Server”](#).
- **KIE Server Keystore Password (KIE_SERVER_HTTPS_PASSWORD)**: The password for the keystore that you created in [Section 2.2, “Creating the secrets for Decision Server”](#).
- **KIE Server Container Deployment (KIE_SERVER_CONTAINER_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.
- **Git Repository URL (SOURCE_REPOSITORY_URL)**: The URL for the Git repository that contains the source for your decision service.
- **Git Reference (SOURCE_REPOSITORY_REF)**: The branch in the Git repository

- **Context Directory (CONTEXT_DIR)**: The path to the source within the project downloaded from the Git repository
 - **Artifact Directory (ARTIFACT_DIR)**: The path within the project that contains the required binary files (KJAR files and any other necessary files) after a successful Maven build. Normally this directory is the target directory of the build. However, you can provide prebuilt binaries in this directory in the Git repository
 - **ImageStream Namespace (IMAGE_STREAM_NAMESPACE)**: The namespace where the image streams are available. If the image streams were already available in your OpenShift environment (see [Section 2.1, “Ensuring the availability of image streams”](#)), the namespace is **openshift**. If you have installed the image streams file, the namespace is the name of the OpenShift project.
3. If your build includes dependencies that are not available on the public Maven tree and require a separate repository, set the parameters to provide this repository:
 - **Maven repository URL (MAVEN_REPO_URL)**: The URL for the Maven repository.
 - **Maven repository username (MAVEN_REPO_USERNAME)**: The username for the Maven repository.
 - **Maven repository password (MAVEN_REPO_PASSWORD)**: The username for the Maven repository.
 4. If you want to use RH-SSO or LDAP authentication, complete the following additional configuration:
 - a. In the RH-SSO or LDAP service, create all user names in the deployment parameters. If you do not set any of the parameters, create users with the default user names. The created users must also be assigned to roles:
 - **KIE_ADMIN_USER**: default user name **adminUser**, roles: **kie-server, rest-all, admin, kiemgmt, Administrators**
 - **KIE_SERVER_USER**: default user name **executionUser**, roles **kie-server, rest-all, guest**
 - b. If you want to configure Red Hat Single Sign On (RH-SSO) authentication, an RH-SSO realm that applies to Red Hat Decision Manager must exist. A client within RH-SSO must also exist for
For the user roles that you can configure in RH-SSO, see [Roles and users](#).

Use one of the following procedures:

- i. If the client for Red Hat Decision Manager within RH-SSO already exists, set the following parameters in the template:
 - **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.
 - **KIE Server RH-SSO Client name (KIE_SERVER_SSO_CLIENT)**: The RH-SSO client name for Decision Server.

- **KIE Server RH-SSO Client Secret (KIE_SERVER_SSO_SECRET)**: The secret string that is set in RH-SSO for the client for Decision Server.
 - **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.
- ii. To create the client for Red Hat Decision Manager within RH-SSO, set the following parameters in the template:
- **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.
 - **KIE Server RH-SSO Client name (KIE_SERVER_SSO_CLIENT)**: The name of the client to create in RH-SSO for Decision Server.
 - **KIE Server RH-SSO Client Secret (KIE_SERVER_SSO_SECRET)**: The secret string to set in RH-SSO for the client for Decision Server.
 - **KIE Server Custom http Route Hostname (KIE_SERVER_HOSTNAME_HTTP)**: The fully qualified host name to use for the HTTP endpoint for Decision Server. If you need to create a client in RH-SSO, you can not leave this parameter blank.
 - **KIE Server Custom https Route Hostname (KIE_SERVER_HOSTNAME_HTTPS)**: The fully qualified host name to use for the HTTPS endpoint for Decision Server. If you need to create a client in RH-SSO, you can not leave this parameter blank.
 - **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.
 - **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.
- c. To configure LDAP, 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](#). Do not configure LDAP authentication and RH-SSO authentication in the same deployment.
5. Complete the creation of the environment, depending on the method that you are using:
- In the OpenShift Web UI, click **Create**.
 - Complete and run the command line.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Friday, October 12, 2018.