



Red Hat Decision Manager 7.3

Deploying a Red Hat Decision Manager
environment on Red Hat OpenShift Container
Platform using Operators

Red Hat Decision Manager 7.3 Deploying a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform using Operators

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

Legal Notice

Copyright © 2019 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, the Red Hat 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 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 7.3 environment on Red Hat OpenShift Container Platform using Operators.

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	5
2.1. ENSURING YOUR ENVIRONMENT IS AUTHENTICATED TO THE RED HAT REGISTRY	5
2.2. CREATING THE SECRETS FOR DECISION SERVER	5
2.3. CREATING THE SECRETS FOR BUSINESS CENTRAL	6
CHAPTER 3. DEPLOYING AND MANAGING A RED HAT DECISION MANAGER ENVIRONMENT USING OPENSIFT OPERATORS	7
3.1. SUBSCRIBING TO THE BUSINESS AUTOMATION OPERATOR	7
3.2. DEPLOYING A RED HAT DECISION MANAGER ENVIRONMENT USING THE OPERATOR	7
3.3. MODIFYING AN ENVIRONMENT THAT IS DEPLOYED USING OPERATORS	11
3.4. PROVIDING THE LDAP ROLE MAPPING FILE	12
APPENDIX A. VERSIONING INFORMATION	14

PREFACE

As a system engineer, you can deploy a Red Hat Decision Manager environment on Red Hat OpenShift Container Platform to provide an infrastructure to develop or execute processes 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.



NOTE

This functionality is currently for Technology Preview only. For more information on Red Hat Technology Preview features, see [Technology Preview Features Scope](#).

Prerequisites

- At least four gigabytes of memory are available in the OpenShift environment.
- The OpenShift Operator Framework is installed and started in the OpenShift environment.
- The OpenShift project for the deployment is created.
- You are logged in to the project using the OpenShift 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 following sizes are required:
 - By default, Business Central requires one 1Gi PV. You can change the PV size for Business Central persistent storage.
- If you intend to scale any of the Business Central pods, your OpenShift environment supports persistent volumes with **ReadWriteMany** mode.



IMPORTANT

ReadWriteMany mode is not supported on OpenShift Online and OpenShift Dedicated.

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.

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

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

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



IMPORTANT

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

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

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 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.

Procedure

1. Determine whether Red Hat OpenShift Container Platform was 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 was configured with the user name and password for Red Hat registry access, no further action is required. Otherwise, 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 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. 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 DECISION SERVER

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

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

Procedure

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

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

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
```

2.3. CREATING THE SECRETS FOR BUSINESS CENTRAL

If you are planning to deploy Business Central in your OpenShift environment, note that this component uses an SSL certificate to provide HTTPS access. 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 for Decision Server.

Procedure

1. Generate an SSL keystore with a private and public key for SSL encryption for Business Central. In a production environment, generate a valid signed certificate that matches the expected URL of the Business Central. Save the keystore in a file named **keystore.jks**. Record the name of the certificate and the password of the keystore file.

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

2. 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
```

CHAPTER 3. DEPLOYING AND MANAGING A RED HAT DECISION MANAGER ENVIRONMENT USING OPENSIFT OPERATORS

To deploy a Red Hat Decision Manager environment using OpenShift Operators, you must provide settings for the environment as a YAML source.

When Red Hat OpenShift Container Platform deploys the environment, it creates a YAML description of the environment, and then ensures that the environment is coherent with the description at all times. You can edit the description to modify the environment.

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. If the operator is not available in the catalog, you must download and install it.

Procedure

1. Enter your project in the OpenShift Web cluster console.
2. In the OpenShift Web console navigation panel, select **Operators** and then **Catalog Sources**.
3. Search for **Business Automation**. If you find it, click **Create Subscription** next to it.
4. If the **Business Automation** entry is not available, complete the following steps:
 - a. Log in to the OpenShift environment as an administrator.
 - b. Enter the following command:

```
$ oc create -f https://raw.githubusercontent.com/kiogroup/ki-cloud-operator/1.0.1/deploy/catalog\_resources/redhat/catalog-source.yaml
```
 - c. Refresh the OpenShift Web console in the browser.
 - d. Search for **Business Automation** again. Click **Create Subscription** next to it.
5. A YAML description of the new subscription is displayed. Click **Create** to create the subscription.

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

After subscribing to the operator, you can *create an application* to deploy an environment.

Procedure

1. Enter your project in the OpenShift Web cluster console.
2. In the OpenShift Web console navigation panel, select **Operators** and then **Subscriptions**.
3. Click the name of the subscription that contains **businessautomation**. Information about this subscription is displayed.

4. Under the **Installed version** heading, click the version name of the subscription. An overview of the subscription is displayed.
5. Click **Create KieApp**. A YAML source is displayed.
6. Set the **name** field to an application name that is unique in the project.
7. Set the **environment** field to the required environment type. Each of the types is a default deployment pattern. You can modify the pattern by editing this YAML source; you can also modify the deployment after it is completed. The following types are available:
 - **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 Decision 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 Decision Server for test execution of the services. You can also use this environment to run services for staging and production purposes. You can add Decision 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 Decision Server for test execution of the services. This version of the authoring environment supports scaling the Business Central pod to ensure high availability.
 - **rhdm-production-immutable**: An alternate environment for running existing services for staging and production purposes. You can configure one or more Decision Server replicated pods that build a service from source. 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.
8. Add lines to the YAML source to modify the environment, using the snippets listed in this document as a reference:
 - The following snippet adds an immutable Decision Server pod that builds a service from source. You must add at least one copy of this snippet when creating an immutable environment.

```
objects:
  servers:
    - 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.
- The following snippet configures the number and settings of Decision Servers that are managed by an existing Business Central in your environment. Six servers, under three different name sets, are included in the snippet.

```

apiVersion: app.kiegroup.org/v1
kind: KieApp
metadata:
  name: server-config
spec:
  environment: <environment_type>
  objects:
    console:
      env:
        - name: MY_VALUE
          value: "example"
    servers:
      # Kieserver sets will be named sequentially server-config-kieserver1, server-config-
      kieserver1-2
      - deployments: 2
        # Env variables that will be added to all the kie servers in this set
        env:
          - name: MY_VALUE
            value: "example"
        # Override default memory limits for all the kie servers in this set
        resources:
          limits:
            memory: 2Gi
      # Kieserver sets will be named sequentially server-config-kieserver2, server-config-
      kieserver2-2
      - deployments: 2
        # Env variables that will be added to all the kie servers in this set
        env:
          - name: MY_VALUE
            value: "example"
      # Kieserver sets will be named sequentially server, server-2
      - name: server
        deployments: 2
        env:
          - name: MY_VALUE
            value: "example"
        # Override default memory limits for all the kie servers in this set
        resources:
          limits:
            memory: 2Gi
    
```

Replace `<environment_type>` with the type of environment that you want to configure.

- The following snippet configures Decision Servers and a Business Central using existing secrets for HTTPS communication, as required for a production environment. In this example, two servers are created with the **server-a-keystore** secret. (For instructions about creating the secrets, see [Section 2.3, "Creating the secrets for Business Central"](#) and [Section 2.2, "Creating the secrets for Decision Server"](#).)

```
apiVersion: app.kiegroup.org/v1
kind: KieApp
metadata:
  name: keystore-config
spec:
  environment: <environment_type>
  objects:
    console:
      keystoreSecret: console-keystore
    servers:
      - name: server-a
        deployments: 2
        keystoreSecret: server-a-keystore
      - name: server-b
        keystoreSecret: server-b-keystore
```

Replace `<environment_type>` with the type of environment that you want to configure.

- The following snippet sets the password for the administrator user (**admin**) and the application name (**app2**):

```
commonConfig:
  adminPassword: password
  applicationName: app2
```

- The following snippet sets up LDAP authentication. The 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](#).

```
auth:
  ldap:
    url: ldaps://myldap.example.com
    bindDN: uid=admin,ou=users,ou=example,ou=com
    bindCredential: s3cret
    baseCtxDN: ou=users,ou=example,ou=com
    baseFilter: (uid={0})
    searchScope: SUBTREE_SCOPE
    roleAttributeID: memberOf
    rolesCtxDN: ou=groups,ou=example,ou=com
    roleFilter: (memberOf={1})
    defaultRole: guest
  roleMapper:
    rolesProperties: /conf/roleMapper.properties
    replaceRole: true
```

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

rolesProperties value to the fully qualified pathname 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 about providing this file, see [Section 3.4, “Providing the LDAP role mapping file”](#).

If the **replaceRole** parameter is set to **true**, mapped roles replace the roles defined on the LDAP server. If the parameter is set to **false**, both mapped roles and roles defined on the LDAP server are set as user application roles. The default setting is **false**.

9. After completing the modification of the YAML source, click **Create** to create the application.



NOTE

You can view other configuration samples, for example, for servers that use different database servers, at <https://github.com/kiegroup/kie-cloud-operator/tree/1.0.1/deploy/examples>

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 pod, 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 Decision Servers, and scale Decision Servers.

Procedure

1. Enter your project in the OpenShift Web cluster console.
2. In the OpenShift Web console navigation panel, select **Operators** and then **Subscriptions**.
3. Click the name of the subscription that contains **businessautomation**. Information about this subscription is displayed.
4. Under the **Installed version** heading, click the version name of the subscription. An overview of the subscription is displayed.
5. Select the **Instances** tab.
6. Click the name of a deployed environment.
7. Select the **YAML** tab. A YAML source is displayed.
8. If you want to change common settings, such as passwords, edit the values under **commonConfig**.
9. If you want to add new Decision 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 Decision 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
10. If you want to scale a Decision 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.
 11. Click **Save Changes**.
 12. Wait for a **This object has been updated** pop-up message.
 13. Click **Reload** to view the new YAML description of the environment.

3.4. PROVIDING THE LDAP ROLE MAPPING FILE

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

Procedure

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

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

For example:

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

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

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


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

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**.

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Wednesday, May 8, 2019.