Red Hat Process Automation Manager 7.8

Deploying a Red Hat Process Automation Manager fixed managed server environment on Red Hat OpenShift Container Platform

Last Updated: 2021-07-01

Red Hat Process Automation Manager 7.8 Deploying a Red Hat Process Automation Manager fixed managed server environment on Red Hat OpenShift Container Platform

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

Legal Notice

Abstract

This document describes how to deploy a Red Hat Process Automation Manager 7.8 fixed managed server environment on Red Hat OpenShift Container Platform.

Table of Contents

PREFACE

As a system engineer, you can deploy a Red Hat Process Automation Manager fixed managed server environment on Red Hat OpenShift Container Platform to provide an infrastructure to execute services, process applications, and other business assets. This environment includes a fixed number of KIE Servers in a single deployment; you cannot add or remove servers at a later point. You can use Business Central Monitoring to manage and update the processes running on KIE Servers in this environment.

Prerequisites

- Red Hat OpenShift Container Platform version 3.11 is deployed.
- At least four gigabytes of memory are available in the OpenShift cluster/namespace.
- The OpenShift project for the deployment is created.
- You are logged in to the project using the **oc** command. For more information about the **oc** command-line tool, see the OpenShift CLI Reference. If you want to use the OpenShift Web console to deploy templates, you must also be logged on using the Web console.
- 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:
    - Each deployed replicated set of KIE Server pods requires, by default, one 1Gi PV for the database. 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 Monitoring requires one 64Mi PV.
    - Smart Router requires one 64Mi PV.
- 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.

NOTE

Since Red Hat Process Automation 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 Process Automation Manager templates with Red Hat OpenShift Container Platform 4.x. To deploy Red Hat Process Automation Manager on Red Hat OpenShift Container Platform 4.x, see the instructions in *Deploying a Red Hat Process Automation Manager environment on Red Hat OpenShift Container Platform using Operators*.

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

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

In this solution, components of Red Hat Process Automation 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 Process Automation Manager are available on OpenShift:

- KIE Server, also known as *Execution Server*, is the infrastructure element that runs decision services, process applications, and other deployable assets (collectively referred to as *services*) . All logic of the services runs on execution servers.
  A database server is normally required for KIE Server. You can provide a database server in another OpenShift pod or configure an execution server on OpenShift to use any other database server. Alternatively, KIE Server can use an H2 database; in this case, you cannot scale the pod.

  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 use the same database server and 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 and monitoring console. You can use Business Central to develop services and deploy them to KIE Servers. You can also use Business Central to monitor the execution of processes.
  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.

- Business Central Monitoring is a web-based management and monitoring console. It can manage the deployment of services to KIE Servers and provide monitoring information, but does not include authoring capabilities. You can use this component to manage staging and production environments.

- Smart Router is an optional layer between KIE Servers and other components that interact with them. When your environment includes many services running on different KIE Servers, Smart Router provides a single endpoint to all client applications. A client application can make a REST API call that requires any service. Smart Router automatically calls the KIE Server that can process a particular request.

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

The following environment types are typical:

- *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. For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform*.

- *Managed deployment*: An environment for running existing services for staging and production purposes. This environment includes several groups of KIE Server pods; you can deploy and undeploy services on every such group and also scale the group up or down as necessary. Use Business Central Monitoring to deploy, run, and stop the services and to monitor their execution. You can deploy two types of managed environment. In a *freeform* server environment, you initially deploy Business Central Monitoring and one KIE Server. You can additionally deploy any number of KIE Servers. Business Central Monitoring can connects to all servers in the same namespace. For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager freeform managed server environment on Red Hat OpenShift Container Platform*.

  Alternatively, you can deploy a *fixed* managed server environment. A single deployment includes Business Central Monitoring, Smart Router, and a preset number of KIE Servers (by default, two servers, but you can modify the template to change the number). You cannot easily add or remove servers at a later time. For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager fixed 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 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. Optionally, you can use Business Central Monitoring to monitor the performance of the environment and to stop and restart some of the service instances, but not to deploy additional services to any KIE Server or undeploy any existing ones (you cannot add or remove containers). For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform*.

You can also deploy a *trial* or evaluation environment. This environment includes Business Central and a 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. For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager trial environment on Red Hat OpenShift Container Platform*.

To deploy a Red Hat Process Automation Manager environment on OpenShift, you can use the templates that are provided with Red Hat Process Automation Manager. You can modify the templates to ensure that the configuration suits your environment.

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

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

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

To deploy Red Hat Process Automation 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 rhpam-businesscentral | grep -F 7.8
   $ oc get imagestreamtag -n openshift | grep -F rhpam-kieserver | grep -F 7.8
   ```

   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 **rhpam-7.8.0-openshift-templates.zip** product deliverable file from the Software Downloads page and extract the **rhpam78-image-streams.yaml** file.

g. Enter the following command:

```
$ oc apply -f rhpam78-image-streams.yaml
```

NOTE

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

2.2. CREATING THE SECRETS FOR 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 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. Save the keystore in a file named **keystore.jks**.

3. Record the name of the certificate. The default value for this name in Red Hat Process Automation Manager configuration is **jboss**.

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

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

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

2.3. CREATING THE SECRETS FOR 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 with a private and public key for SSL encryption for Business Central. 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. Save the keystore in a file named **keystore.jks**.
3. Record the name of the certificate. The default value for this name in Red Hat Process Automation Manager configuration is **jboss**.
4. Record the password of the keystore file. The default value for this name in Red Hat Process Automation Manager configuration is **mykeystorepass**.
5. Use the **oc** command to generate a secret named **businesscentral-app-secret** from the new keystore file:

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

2.4. CREATING THE SECRETS FOR 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 with a private and public key for SSL encryption for Smart Router. 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. Save the keystore in a file named **keystore.jks**.

3. Record the name of the certificate. The default value for this name in Red Hat Process Automation Manager configuration is **jboss**.

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

5. 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
```

2.5. CREATING THE SECRET FOR THE ADMINISTRATIVE USER

You must create a generic secret that contains the user name and password for a Red Hat Process Automation Manager administrative user account. This secret is required for deploying Red Hat Process Automation 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 Process Automation Manager, use the same secret for all these deployments. The components utilize this user account to communicate with each other.

You can also use this user account to log in to Business Central Monitoring.

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 Process Automation 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.

2.6. PROVISIONING PERSISTENT VOLUMES WITH **READWRITEMANY** ACCESS MODE USING NFS

If you want to deploy Business Central Monitoring, 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 *Configuring Clusters* guide in the Red Hat OpenShift Container Platform 3.11 documentation.

2.7. 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 separate mirror repository.

   Alternatively, if you use a custom external repository (for example, Nexus) for your services, you can use the same repository as a mirror repository.

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

   a. Click **Red Hat Process Automation Manager 7.8.0 Offliner Content Listto** download the **rhpam-7.8.0-offliner.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

   b. Extract the contents of the **rhpam-7.8.0-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. FIXED MANAGED SERVER ENVIRONMENT

You can deploy a fixed managed server environment that, in a single deployment, includes several different pods running KIE Server. No processes are initially loaded on the servers. The database servers are, by default, also run in pods. Each KIE Server pod can be separately scaled as necessary.

A pod with Business Central Monitoring and a pod with Smart Router are also deployed. You must use Business Central Monitoring to deploy, load, and unload processes on your KIE Servers. You can also use it to view monitoring information.

Smart Router is a single endpoint that can receive calls from client applications to any of your processes and route each call automatically to the server that runs the process.

By default, the templates create two independent KIE Servers. You can modify the template to change the number of KIE Servers before deployment. You cannot easily add or remove KIE Servers at a later time.

You must provide a Maven repository with the processes (KJAR files) that you want to deploy on the servers. Your integration process must ensure that the required versions of the processes are uploaded to the Maven repository. You can use Business Central in a development environment to create the processes and upload them to the Maven repository.

3.1. DEPLOYING A FIXED MANAGED SERVER ENVIRONMENT

You can deploy a fixed managed server environment using a single template. The name of the template file is **rhpam78-prod.yaml**.

The template includes two KIE Server pods (with PostgreSQL database pods), Smart Router in a high-availability configuration, and Business Central Monitoring in a high-availability configuration.

You can change the number of replicas of all components when configuring the deployment. If you want to modify the number of independent KIE Server pods or to use a different database server, you must modify the template. For instructions about modifying the template, see Section 3.4, “Modifying a template for a managed environment”.

NOTE

The fixed managed environment template is deprecated in Red Hat Process Automation Manager 7.8. It will be removed in a future release.

3.1.1. Starting configuration of the template for a fixed managed server environment

To deploy a fixed managed server environment, use the **rhpam78-prod.yaml** template file.

Procedure

1. Download the **rhpam-7.8.0-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.

2. Extract the **rhpam78-prod.yaml** template file.

3. By default, the template includes two KIE Servers. Each of the serves uses a PostgreSQL database server in a pod. To change the number of KIE Servers or to use a MySQL database server in a pod or an external database server, modify the template as described in Section 3.4, “Modifying a template for a managed environment”.

4. 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 **rhpam78-prod.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>/rhpam78-prod.yaml -p
      BUSINESS_CENTRAL_HTTPS_SECRET=businesscentral-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.

        - Use as many **-p PARAMETER=value** pairs as needed to set the required parameters.

Next steps

Set the parameters for the template. Follow the steps in Section 3.1.2, "Setting required parameters for a fixed managed server environment" to set common parameters. You can view the template file to see descriptions for all parameters.

3.1.2. Setting required parameters for a fixed managed server environment

When configuring the template to deploy a fixed managed server environment, you must set the following parameters in all cases.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for a fixed managed server 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 2.5, "Creating the secret for the administrative user".

    - **Business Central Monitoring Server Keystore Secret Name** (**BUSINESS_CENTRAL_HTTPS_SECRET**): The name of the secret for Business Central, as created in Section 2.3, "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 2.2, "Creating the secrets for KIE Server".

    - **Smart Router Keystore Secret Name** (**KIE_SERVER_ROUTER_HTTPS_SECRET**): The name of the secret for Smart Router, as created in Section 2.4, "Creating the secrets for Smart Router".

    - **Business Central Monitoring Server Certificate Name** (**BUSINESS_CENTRAL_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 2.3, "Creating the secrets for Business Central" .

- Business Central Monitoring Server Keystore Password (**BUSINESS_CENTRAL_HTTPS_PASSWORD**): The password for the keystore that you created in Section 2.3, "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 2.2, "Creating the secrets for KIE Server" .

- KIE Server Keystore Password (**KIE_SERVER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 2.2, "Creating the secrets for KIE Server" .

- Smart Router Certificate Name (**KIE_SERVER_ROUTER_HTTPS_NAME**): The name of the certificate in the keystore that you created in Section 2.4, "Creating the secrets for Smart Router".

- Smart Router Keystore Password (**KIE_SERVER_ROUTER_HTTPS_PASSWORD**): The password for the keystore that you created in Section 2.4, "Creating the secrets for Smart Router".

- 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 Monitoring. 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 Mode(**KIE_SERVER_MODE**): In the **rhpam78-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 2.1, "Ensuring the availability of image streams and the image registry"), the namespace is **openshift**. If you have installed the image streams file, the namespace is the name of the OpenShift project.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, “Completing deployment of the template for a fixed managed server environment”.

3.1.3. Configuring the image stream namespace for a fixed managed server 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 3.1.1, “Starting configuration of the template for a fixed managed server environment”.

Procedure

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

3.1.4. Configuring pod replica numbers for a fixed managed server environment

When configuring the template to deploy a fixed managed server environment, you can set the initial number of replicas for KIE Server, Business Central Monitoring, and Smart Router.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, “Starting configuration of the template for a fixed managed server environment”.

Procedure

To configure the numbers of replicas, set the following parameters:

- **Business Central Monitoring Container Replicas** (**BUSINESS_CENTRAL_MONITORING_CONTAINER_REPLICAS**): The number of replicas that the deployment initially creates for Business Central Monitoring. If you do not want to use a high-availability configuration for Business Central Monitoring, set this number to 1.
- **KIE Server Container Replicas**(**KIE_SERVER_CONTAINER_REPLICAS**): The number of replicas that the deployment initially creates for KIE Server.
- **Smart Router Container Replicas** (**SMART_ROUTER_CONTAINER_REPLICAS**): The number of replicas that the deployment initially creates for Smart Router.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, “Completing deployment of the template for a fixed managed server environment”.

3.1.5. Configuring access to a Maven mirror in an environment without a connection to the public Internet for a fixed managed server environment

When configuring the template to deploy a fixed managed server 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 2.7, “Preparing a Maven mirror repository for offline use” .

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, “Starting configuration of the template for a fixed managed server 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 2.7, “Preparing a Maven mirror repository for offline use” . This URL must be accessible from a pod in your OpenShift environment.
- **Maven mirror of** (**MAVEN_MIRROR_OF**): The value that determines which artifacts are to be retrieved from the mirror. For instructions about setting the **mirrorOf** value, see Mirror Settings in the Apache Maven documentation. The default value is **external:***. With this value, Maven retrieves every required artifact from the mirror and does not query any other repositories.
    - If you configure an external Maven repository (**MAVEN_REPO_URL**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror, for example, **external:*,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.
    - If you configure a built-in Business Central Maven repository (**BUSINESS_CENTRAL_MAVEN_SERVICE**), change **MAVEN_MIRROR_OF** to exclude the artifacts in this repository from the mirror: **external:*,!repo-rhpamcentr**.
    - If you configure both repositories, change **MAVEN_MIRROR_OF** to exclude the artifacts in both repositories from the mirror: **external:*,!repo-rhpamcentr,!repo-custom**. Replace **repo-custom** with the ID that you configured in **MAVEN_REPO_ID**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, “Completing deployment of the template for a fixed managed server environment”.

3.1.6. Setting parameters for RH-SSO authentication for a fixed managed server environment

If you want to use RH-SSO authentication, complete the following additional configuration when configuring the template to deploy a fixed managed server environment.

IMPORTANT

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

Prerequisites

- A realm for Red Hat Process Automation Manager is created in the RH-SSO authentication system.
- User names and passwords for Red Hat Process Automation Manager are created in the RH-SSO authentication system. For a list of the available roles, see Chapter 4, *Red Hat Process Automation 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 2.5, "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 Process Automation 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 Process Automation Manager deployment can create the clients. However, this option provides less detailed control over the environment.
- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for a fixed managed server 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 Process Automation 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 Process Automation Manager within RH-SSO, set the following parameters in the template:
      - **Business Central Monitoring RH-SSO Client name** (**BUSINESS_CENTRAL_SSO_CLIENT**): The RH-SSO client name for Business Central Monitoring.
      - For each KIE Server defined in the template:
        - **KIE Server*n* RH-SSO Client name** (**KIE_SERVER*n*_SSO_CLIENT**): The RH-SSO client name for this KIE Server.
        - **KIE Server*n* RH-SSO Client Secret** (**KIE_SERVER*n*_SSO_SECRET**): The secret string that is set in RH-SSO for the client for this KIE Server.

   b. To create the clients for Red Hat Process Automation Manager within RH-SSO, set the following parameters in the template:
      - For each KIE Server defined in the template:
        - **KIE Server*n* RH-SSO Client name** (**KIE_SERVER*n*_SSO_CLIENT**): The name of the client to create in RH-SSO for this KIE Server.

    - **KIE Server*n* RH-SSO Client Secret** (**KIE_SERVER*n*_SSO_SECRET**): The secret string to set in RH-SSO for the client for this 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 Process Automation Manager. You must provide this user name and password in order to create the required clients.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, "Completing deployment of the template for a fixed managed server environment".

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

3.1.7. Setting parameters for LDAP authentication for a fixed managed server environment

If you want to use LDAP authentication, complete the following additional configuration when configuring the template to deploy a fixed managed server 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 Process Automation Manager in the LDAP system. For a list of the available roles, see Chapter 4, *Red Hat Process Automation 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 2.5, "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 3.1.1, "Starting configuration of the template for a fixed managed server 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 .
   If the LDAP server does not define all the roles required for your deployment, you can map LDAP groups to Red Hat Process Automation Manager roles. To enable LDAP role mapping, set the following parameters:

    - **RoleMapping rolesProperties file path** (**AUTH_ROLE_MAPPER_ROLES_PROPERTIES**): The fully qualified path name of a file that defines role mapping, for example, **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**. You must provide this file and mount it at this path in all applicable deployment configurations; for instructions, see Section 3.3, "(Optional) Providing the LDAP role mapping file" .

- RoleMapping replaceRole property(**AUTH_ROLE_MAPPER_REPLACE_ROLE**): If set to **true**, mapped roles replace the roles defined on the LDAP server; if set to **false**, both mapped roles and roles defined on the LDAP server are set as user application roles. The default setting is **false**.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, "Completing deployment of the template for a fixed managed server environment".

3.1.8. Setting parameters for using an external database server for a fixed managed server environment

If you modified the template to use an external database server for the KIE Server, as described in Section 3.4, "Modifying a template for a managed environment" , complete the following additional configuration when configuring the template to deploy a fixed managed server environment.

Prerequisites

- You started the configuration of the template, as described in Section 3.1.1, "Starting configuration of the template for a fixed managed server environment".

Procedure

1. Set the following parameters:
   - KIE Server External Database Driver(**KIE_SERVER_EXTERNALDB_DRIVER**): The driver for the server, depending on the server type:
     - **mysql**
     - **postgresql**
     - **mariadb**
     - **mssql**
     - **db2**
     - **oracle**
     - **sybase**
   - KIE Server External Database User(**KIE_SERVER_EXTERNALDB_USER**) and KIE Server External Database Password (**KIE_SERVER_EXTERNALDB_PWD**): The user name and password for the external database server
   - KIE Server External Database URL(**KIE_SERVER_EXTERNALDB_URL**): The JDBC URL for the external database server

> NOTE
>
> If you are using the EntrepriseDB Postgres database server, use an URL starting with **jdbc:postgresql://** and not with **jdbc:edb://**. Alternatively, do not set the URL and set the host and port parameters instead.

- KIE Server External Database Host(**KIE_SERVER_EXTERNALDB_SERVICE_HOST**) and KIE Server External Database Port (**KIE_SERVER_EXTERNALDB_SERVICE_PORT**): The host name and port number of the external database server. You can set these parameters as an alternative to setting the **KIE_SERVER_EXTERNALDB_URL** parameter.
- KIE Server External Database Dialect(**KIE_SERVER_EXTERNALDB_DIALECT**): The Hibernate dialect for the server, depending on the server type. The common settings are:
  - **org.hibernate.dialect.MySQL5InnoDBDialect**
  - **org.hibernate.dialect.MySQL8Dialect**
  - **org.hibernate.dialect.MariaDB102Dialect**
  - **org.hibernate.dialect.PostgreSQL95Dialect**
  - **org.hibernate.dialect.PostgresPlusDialect** (used for EntrepriseDB Postgres Advanced Server)
  - **org.hibernate.dialect.SQLServer2012Dialect** (used for MS SQL)
  - **org.hibernate.dialect.DB2Dialect**
  - **org.hibernate.dialect.Oracle10gDialect**
  - **org.hibernate.dialect.SybaseASE15Dialect**
    For a complete list of supported dialects, see Table A.7 in Hibernate properties in the Red Hat JBoss EAP documentation.
- KIE Server External Database name(**KIE_SERVER_EXTERNALDB_DB**): The database name to use on the external database server
- JDBC Connection Checker class (**KIE_SERVER_EXTERNALDB_CONNECTION_CHECKER**): The name of the JDBC connection checker class for the database server. Without this information, a database server connection cannot be restored after it is lost, for example, if the database server is rebooted.
- JDBC Exception Sorter class (**KIE_SERVER_EXTERNALDB_EXCEPTION_SORTER**): The name of the JDBC exception sorter class for the database server. Without this information, a database server connection cannot be restored after it is lost, for example, if the database server is rebooted.

2. If you created a custom image for using an external database server, as described in Section 3.5, “Building a custom KIE Server extension image for an external database”, set the following parameters:
   - Drivers Extension Image (**EXTENSIONS_IMAGE**): The ImageStreamTag definition of the extension image, for example, **jboss-kie-db2-extension-openshift-image:11.1.4.4**

- **Drivers ImageStream Namespace** (**EXTENSIONS_IMAGE_NAMESPACE**): The namespace to which you uploaded the extension image, for example, **openshift** or your project namespace.

3. If you are using a MySQL version 8 external database server, enable the **mysql_native_password** plugin and use it for authentication. For instructions about this pluding, see Native Pluggable Authentication in the *MySQL 8.0 Reference Manual*.
   If you are using a MySQL version 8 image provided by Red Hat on Red Hat OpenShift Container Platform, to enable the plugin, set the **MYSQL_DEFAULT_AUTHENTICATION_PLUGIN** environment variable to **mysql_native_password**.

   If you created users on the MySQL version 8 server before enabling the **mysql_native_password** plugin, you must update the **mysql-user** table after you enable the plugin.

Next steps

If necessary, set additional parameters.

To complete the deployment, follow the procedure in Section 3.1.10, “Completing deployment of the template for a fixed managed server environment”.

3.1.9. Enabling Prometheus metric collection for a fixed managed server 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 3.1.1, “Starting configuration of the template for a fixed managed server 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 3.1.10, “Completing deployment of the template for a fixed managed server environment”.

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

3.1.10. Completing deployment of the template for a fixed managed server 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.

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

Components of your Red Hat Process Automation Manager infrastructure might need to use HTTPS access to servers that have a self-signed HTTPS certificate. For example, Business Central Monitoring 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 Process Automation Manager components to communicate with servers that use self-signed HTTPS server certificates.

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-rhpamcentr --add --overwrite --name=custom-trustore-volume --
mount-path /etc/custom-secret-volume --secret-name=custom-secret

oc set env dc/myapp-rhpamcentr 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.

3.3. (OPTIONAL) PROVIDING THE LDAP ROLE MAPPING FILE

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

Procedure

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

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

For example:

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

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

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

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

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

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

- ***myapp*-rhpamcentrmon**: Business Central Monitoring
- ***myapp*-kieserver-*n***: KIE Server number *n*. By default, the numbers are 1 and 2.

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** .

3.4. MODIFYING A TEMPLATE FOR A MANAGED ENVIRONMENT

To adjust the managed environment to your needs, you need to modify the **rhpam78-prod.yaml** template before deploying the environment.

By default, the templates create two replicated KIE Server pods. You can deploy separate processes on each of the pods. To add more replicated KIE Server pods, you need to modify the template before deploying the environment.

By default, the templates create a PostgreSQL pod to provide the database server for each replicated KIE Server. If you prefer to use PostgreSQL or to use an external server (outside the OpenShift project), you need to modify the template before deploying the environment.

For the **rhpam78-prod.yaml** template you can also adjust the initial number of replicas for Business Central Monitoring.

An OpenShift template defines a set of objects that can be created by OpenShift. To change an environment configuration, you need to modify, add, or delete these objects. To simplify this task, comments are provided in the Red Hat Process Automation Manager templates.

Some comments mark blocks within the template, staring with **BEGIN** and ending with **END**. For example, the following block is named **Sample block**:

```
## Sample block BEGIN
sample line 1
sample line 2
sample line 3
## Sample block END
```

For some changes, you might need to replace a block in one template file with a block from another template file provided with Red Hat Process Automation Manager. In this case, delete the block, then paste the new block in its exact location.

Note that named blocks can be nested.

Procedure

- If you want to add more replicated KIE Server pods, repeat the following actions for every additional pod:

    1. Pick a number for the new pod. The default pods have the numbers **1** and **2**, so you can use **3** for the first new pod, then **4** and so on.
    2. Copy the following blocks of the file, marked with comments from **BEGIN** to **END**, into the end of the file:
        - **KIE server services 1**
        - **PostgreSQL service 1**
        - **KIE server routes 1**
        - **KIE server deployment config 1**
        - **PostgreSQL deployment config 1**
        - **PostgreSQL persistent volume claim 1**
    3. In the new copies, replace all instances of **-1** with the new pod number, for example, **-3**.

- If you want to use MySQL instead of PostgreSQL, replace several blocks of the file, marked with comments from **BEGIN** to **END**, with blocks from the **rhpam78-kieserver-postgresql.yaml** file, then modify some of the newly added blocks:
    1. Replace the block named **MySQL database parameters** with the block named **PostgreSQL database parameters**. (Take this block and all subsequent replacement blocks from the **rhpam78-kieserver-postgresql.yaml** file.)
    Repeat the following actions for every replicated KIE Server pod number, for example, **1** and **2** in the unmodified template. **N** refers to the pod number, for example, **1**.
        - Replace the block named **PostgreSQL service N** with the block named **MySQL service**.
        - Replace the block named **PostgreSQL driver settings N** with the block named **MySQL driver settings**.
        - Replace the block named **PostgreSQL deployment config N** with the block named **MySQL deployment config**.
        - Replace the block named **PostgreSQL persistent volume claim N** with the block named **MySQL persistent volume claim**.
        - In all the newly added blocks, make the following replacements manually, where **N** is the pod number:
            - **-mysql** with **-mysql-N**, *except* in **-mysql-pvol** and in **-mysql-claim**
            - **-mysql-claim** with **-mysql-claim-N**
- If you want to use an external database server, replace several blocks of the file, marked with comments from **BEGIN** to **END**, with blocks from the **rhpam78-kieserver-externaldb.yaml** file, remove some blocks, and modify some of the newly added blocks:
    1. Replace the block named **MySQL database parameters** with the block named **External database parameters**. (Take this block and all subsequent replacement blocks from the **rhpam78-kieserver-external.yaml** file.)
    Repeat the following actions for every replicated KIE Server pod number, for example, **1** and **2** in the unmodified template. **N** refers to the pod number, for example, **1**.

- Remove the block named **PostgreSQL service N**
- Remove the block named **PostgreSQL deployment config N**
- Remove the block named **PostgreSQL persistent volume claim N**
- Replace the block named **PostgreSQL driver settings N** with the block named **External database driver settings**.
- In the new **External database driver settings** block, if any of the following values are different for different KIE Server pods in the infrastructure, set the values for this particular pod:
  - **RHPAM_USERNAME**: The user name for logging in to the database server
  - **RHPAM_PASSWORD**: The password for logging in to the database server
  - **RHPAM_XA_CONNECTION_PROPERTY_URL**: The full URL for logging in to the database server
  - **RHPAM_SERVICE_HOST**: The host name of the database server
  - **RHPAM_DATABASE**: The database name

IMPORTANT

The standard KIE Server image includes drivers for MySQL, MariaDB, and PostgreSQL external database servers. If you want to use another database server, you must build a custom KIE Server image. For instructions, see Section 3.5, “Building a custom KIE Server extension image for an external database”.

- If you want to change the number of replicas initially created for Business Central Monitoring, on the line below the comment **## Replicas for Business Central Monitoring**, change the number of replicas to the desired value.

3.5. BUILDING A CUSTOM KIE SERVER EXTENSION IMAGE FOR AN EXTERNAL DATABASE

If you want to use an external database server for a KIE Server and the database server is not a MySQL or PostgreSQL server, you must build a custom KIE Server extension image with drivers for this server before deploying your environment.

Complete the steps in this build procedure to provide drivers for any of the following database servers:

- Microsoft SQL Server
- IBM DB2
- Oracle Database
- Sybase

Optionally, you can use this procedure to build a new version of drivers for any of the following database servers:

- MySQL

- MariaDB
- PostgreSQL

For the supported versions of the database servers, see Red Hat Process Automation Manager 7 Supported Configurations.

The build procedure creates a custom extension image that extends the existing KIE Server image. You must import this custom extension image into your OpenShift environment and then reference it in the **EXTENSIONS_IMAGE** parameter.

Prerequisites

- You are logged in to your OpenShift environment using the **oc** command. Your OpenShift user must have the **registry-editor** role.
- For Oracle Database, IBM DB2, or Sybase, you downloaded the JDBC driver from the database server vendor.
- You have installed the following required software:
    - Docker: For installation instructions, see Get Docker.
    - Cekit version 3.2: For installation instructions, see Installation.
    - The following libraries and extensions for Cekit. For more information, see Dependencies.
        - **docker**, provided by the **python3-docker** package or similar package
        - **docker-squash**, provided by the **python3-docker-squash** package or similar package
        - **behave**, provided by the **python3-behave** package or similar package

Procedure

1. For IBM DB2, Oracle Database, or Sybase, provide the JDBC driver JAR file in a local directory.
2. Download the **rhpam-7.8.0-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat Customer Portal.
3. Unzip the file and, using the command line, change to the **templates/contrib/jdbc/cekit** directory of the unzipped file. This directory contains the source code for the custom build.
4. Enter one of the following commands, depending on the database server type:
    - For Microsoft SQL Server:

      ```
      make mssql
      ```

    - For MySQL:

      ```
      make mysql
      ```

    - For PostgreSQL:

      ```
      make postgresql
      ```

- For MariaDB:

  ```
  make mariadb
  ```

- For IBM DB2:

  ```
  make db2 artifact=/tmp/db2jcc4.jar version=10.2
  ```

  In this command, replace **/tmp/db2jcc4.jar** with the path name of the IBM DB2 driver and **10.2** with the version of the driver.

- For Oracle Database:

  ```
  make oracle artifact=/tmp/ojdbc7.jar version=7.0
  ```

  In this command, replace **/tmp/ojdbc7.jar** with the path name of the Oracle Database driver and **7.0** with the version of the driver.

- For Sybase:

  ```
  make build sybase artifact=/tmp/jconn4-16.0_PL05.jar version=16.0_PL05
  ```

  In this command, replace **/tmp/jconn4-16.0_PL05.jar** with the path name of the downloaded Sybase driver and **16.0_PL05** with the version of the driver.

  Alternatively, if you need to update the driver class or driver XA class for the Sybase driver, you can set the **DRIVER_CLASS** or **DRIVER_XA_CLASS** variable for this command, for example:

  ```
  export DRIVER_CLASS=another.class.Sybase && make sybase artifact=/tmp/jconn4-
  16.0_PL05.jar version=16.0_PL05
  ```

5. Enter the following command to list the Docker images that are available locally:

   ```
   docker images
   ```

   Note the name of the image that was built, for example, **jboss-kie-db2-extension-openshift-image**, and the version tag of the image, for example, **11.1.4.4** (not the **latest** tag).

6. Access the registry of your OpenShift environment directly and push the image to the registry. Depending on your user permissions, you can push the image into the **openshift** namespace or into a project namespace. For instructions about accessing the registry and pushing the images, see Accessing the Registry Directly in the Red Hat OpenShift Container Platform product documentation.

7. When configuring your KIE Server deployment with a template that supports an external database server, set the following parameters:

   - **Drivers Extension Image** (**EXTENSIONS_IMAGE**): The ImageStreamTag definition of the extension image, for example, **jboss-kie-db2-extension-openshift-image:11.1.4.4**

   - **Drivers ImageStream Namespace** (**EXTENSIONS_IMAGE_NAMESPACE**): The namespace to which you uploaded the extension image, for example, **openshift** or your project namespace.

CHAPTER 4. RED HAT PROCESS AUTOMATION 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 Process Automation Manager user roles.

NOTE

The **admin**, **analyst**, **developer**, **manager**, **process-admin**, **user**, 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 Process Automation Manager.

- **analyst**: Users with the **analyst** role have access to all high-level features. They can model and execute their 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.

- **developer**: Users with the **developer** role have access to almost all features and can manage rules, models, process flows, forms, and dashboards. They can manage the asset repository, they can create, build, and deploy projects, and they can use Red Hat CodeReady Studio to view processes. Only certain administrative functions such as creating and cloning a new repository are hidden from users with the **developer** role.

- **manager**: Users with the **manager** role can view reports. These users are usually interested in statistics about the business processes and their performance, business indicators, and other business-related reporting. A user with this role has access only to process and task reports.

- **process-admin**: Users with the **process-admin** role are business process administrators. They have full access to business processes, business tasks, and execution errors. These users can also view business reports and have access to the Task Inbox list.

- **user**: Users with the **user** role can work on the Task Inbox list, which contains business tasks that are part of currently running processes. Users with this role can view process and task reports and manage processes.

- **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. This role is mandatory for users to have access to **Manage** and **Track** views in Business Central.

CHAPTER 5. OPENSHIFT TEMPLATE REFERENCE INFORMATION

Red Hat Process Automation Manager provides the following OpenShift templates. To access the templates, download and extract the **rhpam-7.8.0-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat customer portal.

- **rhpam78-prod.yaml** provides a high-availability Business Central Monitoring instance, a Smart Router, two distinct KIE Servers connected to the Business Central and to the Smart Router, and two PostgreSQL instances. Each KIE Server uses its own PostgreSQL instance. You can use this environment to execute business assets in a production or staging environment. You can configure the number of replicas for each component. For details about this template, see Section 5.1, “rhpam78-prod.yaml template”.

5.1. RHPAM78-PROD.YAML TEMPLATE

Application template for a managed HA production runtime environment, for Red Hat Process Automation Manager 7.8 - Deprecated

5.1.1. 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
MAVEN_MIRROR_URL	**MAVEN_MIRROR_URL**	Maven mirror that the 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

Variable name	Image Environment Variable	Description	Example value	Required
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-rhpamcentr,!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/	True
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
BUSINESS_CENTRAL_MAVEN_SERVICE	**RHPAMCENTR_MAVEN_REPO_SERVICE**	The service name for the optional Business Central, where it can be reached, to allow service lookups (for maven repo usage), if required.	myapp-rhpamcentr	False

Variable name	Image Environment Variable	Description	Example value	Required
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 Process Automation Manager images are installed. These ImageStreams are normally installed in the openshift namespace. You need to modify this parameter only if you installed the ImageStream in a different namespace/project. Default is "openshift".	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	The name of the image stream to use for KIE server. Default is "rhpam-kieserver-rhel8".	rhpam-kieserver-rhel8	True
IMAGE_STREAM_TAG	–	A named pointer to an image in an image stream. Default is "7.8.0".	7.8.0	True
SMART_ROUTER_HOSTNAME_HTTP	–	Custom hostname for http service route. Leave blank for default hostname, e.g. <application-name>-smartrouter-<project>.<default-domain-suffix>'	–	False

Variable name	Image Environment Variable	Description	Example value	Required
SMART_ROUTER_HOSTNAME_HTTPS	–	Custom hostname for https service route. Leave blank for default hostname, e.g. secure-<application-name>-smartrouter-<project>.<default-domain-suffix>'	–	False
KIE_SERVER_ROUTER_ID	**KIE_SERVER_ROUTER_ID**	Router ID used when connecting to the controller. (router property org.kie.server.router.id)	kie-server-router	True
KIE_SERVER_ROUTER_PROTOCOL	**KIE_SERVER_ROUTER_PROTOCOL**	KIE server router protocol. (Used to build the org.kie.server.router.url.external property)	http	False
KIE_SERVER_ROUTER_URL_EXTERNAL	**KIE_SERVER_ROUTER_URL_EXTERNAL**	Public URL where the router can be found. Format http://<host>:<port> (router property org.kie.server.router.url.external)	–	False
KIE_SERVER_ROUTER_NAME	**KIE_SERVER_ROUTER_NAME**	Router name used when connecting to the controller. (router property org.kie.server.router.name)	KIE Server Router	True

Variable name	Image Environment Variable	Description	Example value	Required
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_PERSISTENCE_DS	**KIE_SERVER_PERSISTENCE_DS**	KIE server persistence datasource. (Sets the org.kie.server.persistence.ds system property)	java:/jboss/datasources/rhpam	False
POSTGRESQL_IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStream for the PostgreSQL image is installed. The ImageStream is already installed in the openshift namespace. You need to modify this parameter only if you installed the ImageStream in a different namespace/project. Default is "openshift".	openshift	False
POSTGRESQL_IMAGE_STREAM_TAG	–	The PostgreSQL image version, which is intended to correspond to the PostgreSQL version. Default is "10".	10	False
KIE_SERVER_POSTGRESQL_USER	**RHPAM_USERNAME**	KIE server PostgreSQL database user name.	rhpam	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_POSTGRESQL_PWD	**RHPAM_PASSWORD**	KIE server PostgreSQL database password.	–	False
KIE_SERVER_POSTGRESQL_DB	**RHPAM_DATABASE**	KIE server PostgreSQL database name.	rhpam7	False
POSTGRESQL_MAX_PREPARED_TRANSACTIONS	**POSTGRESQL_MAX_PREPARED_TRANSACTIONS**	Allows the PostgreSQL to handle XA transactions.	100	True
DB_VOLUME_CAPACITY	–	Size of persistent storage for the database volume.	1Gi	True
KIE_SERVER_POSTGRESQL_DIALECT	**KIE_SERVER_PERSISTENCE_DIALECT**	KIE server PostgreSQL Hibernate dialect.	org.hibernate.dialect.PostgreSQLDialect	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

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
BUSINESS_CENTRAL_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-rhpamcentrmon-<project>.<default-domain-suffix>	–	False
BUSINESS_CENTRAL_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-rhpamcentrmon-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER1_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER1_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER1_USE_SECURE_ROUTE_NAME	**KIE_SERVER_USE_SECURE_ROUTE_NAME**	If true, the KIE server will use secure-<application-name>-kieserver vs. <application-name>-kieserver as the KIE server route endpoint for Business Central to report. Therefore, Business Central displays the secure link to the user.	false	False
KIE_SERVER2_HOSTNAME_HTTP	**HOSTNAME_HTTP**	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	–	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER2_HOSTNAME_HTTPS	**HOSTNAME_HTTPS**	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-kieserver-<project>.<default-domain-suffix>	–	False
KIE_SERVER2_USE_SECURE_ROUTE_NAME	**KIE_SERVER_USE_SECURE_ROUTE_NAME**	If true, will use secure-APPLICATION_NAME-kieserver-2 vs. APPLICATION_NAME-kieserver-2 as the route name.	false	False
BUSINESS_CENTRAL_HTTPS_SECRET	–	The name of the secret containing the keystore file for Business Central.	businesscentral-app-secret	True
BUSINESS_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret.	keystore.jks	False
BUSINESS_CENTRAL_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate.	jboss	False
BUSINESS_CENTRAL_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate.	mykeystorepass	False
KIE_SERVER_ROUTER_HTTPS_SECRET	–	The name of the secret containing the keystore file for Smart Router.	smartrouter-app-secret	True
KIE_SERVER_ROUTER_HTTPS_KEYSTORE	–	The name of the keystore file within the secret.	keystore.jks	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_ROUTER_HTTPS_NAME	**KIE_SERVER_ROUTER_TLS_KEYSTORE_KEYALIAS**	The name associated with the server certificate.	jboss	False
KIE_SERVER_ROUTER_HTTPS_PASSWORD	**KIE_SERVER_ROUTER_TLS_KEYSTORE_PASSWORD**	The password for the keystore and certificate.	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.	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
TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	**TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL**	Sets refresh-interval for the EJB timer service database-data-store.	30000	False

Variable name	Image Environment Variable	Description	Example value	Required
BUSINESS_CENTRAL_MEMORY_LIMIT	–	Business Central Monitoring Container memory limit.	2Gi	False
KIE_SERVER_MEMORY_LIMIT	–	KIE server Container memory limit.	1Gi	False
SMART_ROUTER_MEMORY_LIMIT	–	Smart Router Container memory limit	512Mi	False
BUSINESS_CENTRAL_MONITORING_CONTAINER_REPLICAS	–	Business Central Monitoring Container Replicas, defines how many Business Central Monitoring containers will be started.	3	True
SMART_ROUTER_CONTAINER_REPLICAS	–	Smart Router Container Replicas, defines how many smart router containers will be started.	2	True
KIE_SERVER_CONTAINER_REPLICAS	–	KIE Server Container Replicas, defines how many KIE Server containers will be started.	3	True
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
BUSINESS_CENTRAL_SSO_CLIENT	**SSO_CLIENT**	Business Central Monitoring RH-SSO Client name.	–	False
BUSINESS_CENTRAL_SSO_SECRET	**SSO_SECRET**	Business Central Monitoring RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER1_SSO_CLIENT	**SSO_CLIENT**	KIE Server 1 RH-SSO Client name.	–	False
KIE_SERVER1_SSO_SECRET	**SSO_SECRET**	KIE Server 1 RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	False
KIE_SERVER2_SSO_CLIENT	**SSO_CLIENT**	KIE Server 2 RH-SSO Client name.	–	False
KIE_SERVER2_SSO_SECRET	**SSO_SECRET**	KIE Server 2 RH-SSO Client Secret.	252793ed-7118-4ca8-8dab-5622fa97d892	False
SSO_USERNAME	**SSO_USERNAME**	RH-SSO Realm admin user name for creating 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.	ldap://myldap.example.com	False

Variable name	Image Environment Variable	Description	Example value	Required
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
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

5.1.2. Objects

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

5.1.2.1. Services

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

Service	Port	Name	Description
${APPLICATION_NAME}-rhpamcentrmon	8080	http	All the Business Central Monitoring web server's ports.
	8443	https	
${APPLICATION_NAME}-rhpamcentrmon-ping	8888	ping	The JGroups ping port for clustering.
${APPLICATION_NAME}-smartrouter	9000	http	The smart router server http and https ports.
	9443	https	
${APPLICATION_NAME}-kieserver-1	8080	http	All the KIE server web server's ports. (First KIE server)
	8443	https	
${APPLICATION_NAME}-kieserver-1-ping	8888	ping	The JGroups ping port for clustering.
${APPLICATION_NAME}-kieserver-2	8080	http	All the KIE server web server's ports. (Second KIE server)
	8443	https	
${APPLICATION_NAME}-kieserver-2-ping	8888	ping	The JGroups ping port for clustering.
${APPLICATION_NAME}-postgresql-1	5432	–	The first database server's port.
${APPLICATION_NAME}-postgresql-2	5432	–	The second database server's port.

5.1.2.2. Routes

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

Service	Security	Hostname
${APPLICATION_NAME}-rhpamcentrmon-http	none	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**
${APPLICATION_NAME}-rhpamcentrmon-https	TLS passthrough	**${BUSINESS_CENTRAL_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-kieserver-1-http	none	**${KIE_SERVER1_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-1-https	TLS passthrough	**${KIE_SERVER1_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-kieserver-2-http	none	**${KIE_SERVER2_HOSTNAME_HTTP}**
${APPLICATION_NAME}-kieserver-2-https	TLS passthrough	**${KIE_SERVER2_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-smartrouter-http	none	**${SMART_ROUTER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-smartrouter-https	TLS passthrough	**${SMART_ROUTER_HOSTNAME_HTTPS}**

5.1.2.3. 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.

5.1.2.3.1. 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}-rhpamcentrmon	ImageChange
${APPLICATION_NAME}-smartrouter	ImageChange
${APPLICATION_NAME}-kieserver-1	ImageChange
${APPLICATION_NAME}-postgresql-1	ImageChange

Deployment	Triggers
${APPLICATION_NAME}-kieserver-2	ImageChange
${APPLICATION_NAME}-postgresql-2	ImageChange

5.1.2.3.2. 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}-rhpamcentrmon	3
${APPLICATION_NAME}-smartrouter	2
${APPLICATION_NAME}-kieserver-1	3
${APPLICATION_NAME}-postgresql-1	1
${APPLICATION_NAME}-kieserver-2	3
${APPLICATION_NAME}-postgresql-2	1

5.1.2.3.3. Pod Template

5.1.2.3.3.1. 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
${APPLICATION_NAME}-smartrouter	**${APPLICATION_NAME}-smartrouter**
${APPLICATION_NAME}-kieserver-1	**${APPLICATION_NAME}-kieserver**
${APPLICATION_NAME}-kieserver-2	**${APPLICATION_NAME}-kieserver**

5.1.2.3.3.2. Image

Deployment	Image
${APPLICATION_NAME}-rhpamcentrmon	rhpam-businesscentral-monitoring-rhel8
${APPLICATION_NAME}-smartrouter	rhpam-smartrouter-rhel8
${APPLICATION_NAME}-kieserver-1	**${KIE_SERVER_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-postgresql-1	postgresql
${APPLICATION_NAME}-kieserver-2	**${KIE_SERVER_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-postgresql-2	postgresql

5.1.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhpamcentrmon

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

${APPLICATION_NAME}-kieserver-1

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

${APPLICATION_NAME}-postgresql-1

```
/usr/libexec/check-container
```

${APPLICATION_NAME}-kieserver-2

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

${APPLICATION_NAME}-postgresql-2

```
/usr/libexec/check-container
```

5.1.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhpamcentrmon

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

${APPLICATION_NAME}-kieserver-1

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

${APPLICATION_NAME}-postgresql-1

```
/usr/libexec/check-container --live
```

${APPLICATION_NAME}-kieserver-2

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

${APPLICATION_NAME}-postgresql-2

```
/usr/libexec/check-container --live
```

5.1.2.3.3.5. Exposed Ports

Deployments	Name	Port	Protocol
${APPLICATION_NAME}-rhpamcentrmon	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-smartrouter	http	9000	**TCP**
${APPLICATION_NAME}-kieserver-1	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-postgresql-1	–	5432	**TCP**
${APPLICATION_NAME}-kieserver-2	jolokia	8778	**TCP**
	http	8080	**TCP**
	https	8443	**TCP**
	ping	8888	**TCP**
${APPLICATION_NAME}-postgresql-2	–	5432	**TCP**

5.1.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-rhpamcentrmon	**APPLICATION_USERS_PROPERTIES**	–	**/opt/kie/data/configuration/application-users.properties**
	APPLICATION_ROLES_PROPERTIES	–	**/opt/kie/data/configuration/application-roles.properties**
	KIE_ADMIN_USER	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	MAVEN_MIRROR_URL	Maven mirror that the KIE server must use. If you configure a mirror, this mirror must contain all artifacts that are required for deploying your services.	**${MAVEN_MIRROR_URL}**
	MAVEN_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-rhpamcentr,!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}**

Deployment	Variable name	Description	Example value
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	HTTPS_KEYSTORE_DIR	–	**/etc/businesscentral-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret.	**${BUSINESS_CENTRAL_HTTPS_KEYSTORE}**
	HTTPS_NAME	The name associated with the server certificate.	**${BUSINESS_CENTRAL_HTTPS_NAME}**
	HTTPS_PASSWORD	The password for the keystore and certificate.	**${BUSINESS_CENTRAL_HTTPS_PASSWORD}**
	JGROUPS_PING_PROTOCOL	–	openshift.DNS_PING
	OPENSHIFT_DNS_PING_SERVICE_NAME	–	**${APPLICATION_NAME}-rhpamcentrmon-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	Business Central Monitoring RH-SSO Client Secret.	**${BUSINESS_CENTRAL_SSO_SECRET}**

Deployment	Variable name	Description	Example value
	SSO_CLIENT	Business Central Monitoring RH-SSO Client name.	**${BUSINESS_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name for creating the Client if it doesn’t exist.	**${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.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-rhpamcentrmon-<project>.<default-domain-suffix>	**${BUSINESS_CENTRAL_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-rhpamcentrmon-<project>.<default-domain-suffix>	**${BUSINESS_CENTRAL_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication.	**${AUTH_LDAP_BIND_CREDENTIAL}**

Deployment	Variable name	Description	Example value
	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}**
	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}**
${APPLICATION_NAME}-smartrouter	**KIE_ADMIN_USER**	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_SERVER_ROUTER_HOST	–	–
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PORT_TLS	–	9443
	KIE_SERVER_ROUTER_URL_EXTERNAL	Public URL where the router can be found. Format http://<host>:<port> (router property org.kie.server.router.url.external)	**${KIE_SERVER_ROUTER_URL_EXTERNAL}**
	KIE_SERVER_ROUTER_ID	Router ID used when connecting to the controller. (router property org.kie.server.router.id)	**${KIE_SERVER_ROUTER_ID}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_ROUTER_NAME	Router name used when connecting to the controller. (router property org.kie.server.router.name)	**${KIE_SERVER_ROUTER_NAME}**
	KIE_SERVER_ROUTER_ROUTE_NAME	–	**${APPLICATION_NAME}-smartrouter**
	KIE_SERVER_ROUTER_SERVICE	–	**${APPLICATION_NAME}-smartrouter**
	KIE_SERVER_ROUTER_PROTOCOL	KIE server router protocol. (Used to build the org.kie.server.router.url.external property)	**${KIE_SERVER_ROUTER_PROTOCOL}**
	KIE_SERVER_ROUTER_TLS_KEYSTORE_KEYALIAS	The name associated with the server certificate.	**${KIE_SERVER_ROUTER_HTTPS_NAME}**
	KIE_SERVER_ROUTER_TLS_KEYSTORE_PASSWORD	The password for the keystore and certificate.	**${KIE_SERVER_ROUTER_HTTPS_PASSWORD}**
	KIE_SERVER_ROUTER_TLS_KEYSTORE	–	**/etc/smartrouter-secret-volume/${KIE_SERVER_ROUTER_HTTPS_KEYSTORE}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhpamcentrmon**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	http
	KIE_SERVER_ROUTER_REPO	–	**/opt/rhpam-smartrouter/data**

Deployment	Variable name	Description	Example value
	KIE_SERVER_ROUTER_CONFIG_WATCHER_ENABLED	–	true
${APPLICATION_NAME}-kieserver-1	**KIE_ADMIN_USER**	Admin user name	Set according to the credentials secret
	KIE_ADMIN_PWD	Admin user password	Set according to the credentials secret
	KIE_SERVER_MODE	The KIE Server mode. Valid values are 'DEVELOPMENT' or 'PRODUCTION'. In production mode, you can not deploy SNAPSHOT versions of artifacts on the KIE server and can not change the version of an artifact in an existing container. (Sets the org.kie.server.mode system property).	**${KIE_SERVER_MODE}**
	KIE_MBEANS	KIE server mbeans enabled/disabled. (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	DROOLS_SERVER_FILTER_CLASSES	KIE server class filtering. (Sets the org.drools.server.filter.classes system property)	**${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}**

Deployment	Variable name	Description	Example value
	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)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhpamcentrmon**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	**${APPLICATION_NAME}-kieserver-1**
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver-1**
	KIE_SERVER_USE_SECURE_ROUTE_NAME	If true, the KIE server will use secure-<application-name>-kieserver vs. <application-name>-kieserver as the KIE server route endpoint for Business Central to report. Therefore, Business Central displays the secure link to the user.	**${KIE_SERVER1_USE_SECURE_ROUTE_NAME}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	–	

Deployment	Variable name	Description	Example value
	MAVEN_MIRROR_URL	Maven mirror that the 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	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_ID	–	repo-rhpamcentr
	RHPAMCENTR_MAVEN_REPO_SERVICE	The service name for the optional Business Central, where it can be reached, to allow service lookups (for maven repo usage), if required.	**${BUSINESS_CENTRAL_MAVEN_SERVICE}**
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository. If set, it can be excluded from the optionally configured mirror by adding it to MAVEN_MIRROR_OF. For example: external:*,!repo-rhpamcentr,!repo-custom. If MAVEN_MIRROR_URL is set but MAVEN_MIRROR_ID is not set, an id will be generated randomly, but won't be usable in MAVEN_MIRROR_OF.	**${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}**
	KIE_SERVER_ROUTER_SERVICE	–	**${APPLICATION_NAME}-smartrouter**
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PROTOCOL	KIE server router protocol. (Used to build the org.kie.server.router.url.external property)	**${KIE_SERVER_ROUTER_PROTOCOL}**
	KIE_SERVER_PERSISTENCE_DS	KIE server persistence datasource. (Sets the org.kie.server.persistence.ds system property)	**${KIE_SERVER_PERSISTENCE_DS}**
	DATASOURCES	–	**RHPAM**
	RHPAM_JNDI	KIE server persistence datasource. (Sets the org.kie.server.persistence.ds system property)	**${KIE_SERVER_PERSISTENCE_DS}**
	RHPAM_JTA	–	true
	RHPAM_DATABASE	KIE server PostgreSQL database name.	**${KIE_SERVER_POSTGRESQL_DB}**
	RHPAM_DRIVER	–	postgresql
	KIE_SERVER_PERSISTENCE_DIALECT	KIE server PostgreSQL Hibernate dialect.	**${KIE_SERVER_POSTGRESQL_DIALECT}**
	RHPAM_USERNAME	KIE server PostgreSQL database user name.	**${KIE_SERVER_POSTGRESQL_USER}**

Deployment	Variable name	Description	Example value
	RHPAM_PASSWORD	KIE server PostgreSQL database password.	**${KIE_SERVER_POSTGRESQL_PWD}**
	RHPAM_SERVICE_HOST	–	**${APPLICATION_NAME}-postgresql-1**
	RHPAM_SERVICE_PORT	–	5432
	TIMER_SERVICE_DATA_STORE	–	**${APPLICATION_NAME}-postgresql-1**
	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	Sets refresh-interval for the EJB timer service database-data-store.	**${TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL}**
	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-1-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL.	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**

Deployment	Variable name	Description	Example value
	SSO_SECRET	KIE Server 1 RH-SSO Client Secret.	**${KIE_SERVER1_SSO_SECRET}**
	SSO_CLIENT	KIE Server 1 RH-SSO Client name.	**${KIE_SERVER1_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name for creating 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.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER1_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER1_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication.	**${AUTH_LDAP_BIND_CREDENTIAL}**

Deployment	Variable name	Description	Example value
	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}**
	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}**
${APPLICATION_NAME}-postgresql-1	**POSTGRESQL_USER**	KIE server PostgreSQL database user name.	**${KIE_SERVER_POSTGRESQL_USER}**
	POSTGRESQL_PASSWORD	KIE server PostgreSQL database password.	**${KIE_SERVER_POSTGRESQL_PWD}**
	POSTGRESQL_DATABASE	KIE server PostgreSQL database name.	**${KIE_SERVER_POSTGRESQL_DB}**
	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	Allows the PostgreSQL to handle XA transactions.	**${POSTGRESQL_MAX_PREPARED_TRANSACTIONS}**
${APPLICATION_NAME}-kieserver-2	**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)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_TOKEN	KIE server controller token for bearer authentication. (Sets the org.kie.server.controller.token system property)	**${KIE_SERVER_CONTROLLER_TOKEN}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_CONTROLLER_SERVICE	–	**${APPLICATION_NAME}-rhpamcentrmon**
	KIE_SERVER_CONTROLLER_PROTOCOL	–	ws
	KIE_SERVER_ID	–	**${APPLICATION_NAME}-kieserver-2**
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver-2**
	KIE_SERVER_USE_SECURE_ROUTE_NAME	If true, will use secure-APPLICATION_NAME-kieserver-2 vs. APPLICATION_NAME-kieserver-2 as the route name.	**${KIE_SERVER2_USE_SECURE_ROUTE_NAME}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	–	
	MAVEN_MIRROR_URL	Maven mirror that the 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	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_ID	–	repo-rhpamcentr
	RHPAMCENTR_MAVEN_REPO_SERVICE	The service name for the optional Business Central, where it can be reached, to allow service lookups (for maven repo usage), if required.	**${BUSINESS_CENTRAL_MAVEN_SERVICE}**

Deployment	Variable name	Description	Example value
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	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-rhpamcentr,!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}**
	KIE_SERVER_ROUTER_SERVICE	–	**${APPLICATION_NAME}-smartrouter**
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PROTOCOL	KIE server router protocol. (Used to build the org.kie.server.router.url.external property)	**${KIE_SERVER_ROUTER_PROTOCOL}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_PERSISTENCE_DS	KIE server persistence datasource. (Sets the org.kie.server.persistence.ds system property)	**${KIE_SERVER_PERSISTENCE_DS}**
	DATASOURCES	–	**RHPAM**
	RHPAM_JNDI	KIE server persistence datasource. (Sets the org.kie.server.persistence.ds system property)	**${KIE_SERVER_PERSISTENCE_DS}**
	RHPAM_JTA	–	true
	RHPAM_DATABASE	KIE server PostgreSQL database name.	**${KIE_SERVER_POSTGRESQL_DB}**
	RHPAM_DRIVER	–	postgresql
	KIE_SERVER_PERSISTENCE_DIALECT	KIE server PostgreSQL Hibernate dialect.	**${KIE_SERVER_POSTGRESQL_DIALECT}**
	RHPAM_USERNAME	KIE server PostgreSQL database user name.	**${KIE_SERVER_POSTGRESQL_USER}**
	RHPAM_PASSWORD	KIE server PostgreSQL database password.	**${KIE_SERVER_POSTGRESQL_PWD}**
	RHPAM_SERVICE_HOST	–	**${APPLICATION_NAME}-postgresql-2**
	RHPAM_SERVICE_PORT	–	5432
	TIMER_SERVICE_DATA_STORE	–	**${APPLICATION_NAME}-postgresql-2**
	TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL	Sets refresh-interval for the EJB timer service database-data-store.	**${TIMER_SERVICE_DATA_STORE_REFRESH_INTERVAL}**
	HTTPS_KEYSTORE_DIR	–	**/etc/kieserver-secret-volume**
	HTTPS_KEYSTORE	The name of the keystore file within the secret.	**${KIE_SERVER_HTTPS_KEYSTORE}**

Deployment	Variable name	Description	Example value
	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	–	**${APPLICATION_NAME}-kieserver-2-ping**
	OPENSHIFT_DNS_PING_SERVICE_PORT	–	8888
	SSO_URL	RH-SSO URL.	**${SSO_URL}**
	SSO_OPENIDCONNECT_DEPLOYMENTS	–	ROOT.war
	SSO_REALM	RH-SSO Realm name.	**${SSO_REALM}**
	SSO_SECRET	KIE Server 2 RH-SSO Client Secret.	**${KIE_SERVER2_SSO_SECRET}**
	SSO_CLIENT	KIE Server 2 RH-SSO Client name.	**${KIE_SERVER2_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm admin user name for creating the Client if it doesn’t exist.	**${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}**

Deployment	Variable name	Description	Example value
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as user name.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER2_HOSTNAME_HTTP}**
	HOSTNAME_HTTPS	Custom hostname for https service route. Leave blank for default hostname, e.g.: secure-<application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER2_HOSTNAME_HTTPS}**
	AUTH_LDAP_URL	LDAP Endpoint to connect for authentication.	**${AUTH_LDAP_URL}**
	AUTH_LDAP_BIND_DN	Bind DN used for authentication.	**${AUTH_LDAP_BIND_DN}**
	AUTH_LDAP_BIND_CREDENTIAL	LDAP Credentials used for authentication.	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	**${AUTH_LDAP_BASE_CTX_DN}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry’s DN is used.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the 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}**
${APPLICATION_NAME}-postgresql-2	**POSTGRESQL_USER**	KIE server PostgreSQL database user name.	**${KIE_SERVER_POSTGRESQL_USER}**
	POSTGRESQL_PASSWORD	KIE server PostgreSQL database password.	**${KIE_SERVER_POSTGRESQL_PWD}**
	POSTGRESQL_DATABASE	KIE server PostgreSQL database name.	**${KIE_SERVER_POSTGRESQL_DB}**
	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	Allows the PostgreSQL to handle XA transactions.	**${POSTGRESQL_MAX_PREPARED_TRANSACTIONS}**

5.1.2.3.3.7. Volumes

Deployment	Name	mountPath	Purpose	readOnly
${APPLICATION_NAME}-rhpamcentrmon	businesscentral-keystore-volume	**/etc/businesscentral-secret-volume**	ssl certs	True
${APPLICATION_NAME}-smartrouter	**${APPLICATION_NAME}-smartrouter**	**/opt/rhpam-smartrouter/data**	–	false

Deployment	Name	mountPath	Purpose	readOnly
${APPLICATION_NAME}-kieserver-1	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True
${APPLICATION_NAME}-postgresql-1	**${APPLICATION_NAME}-postgresql-pvol**	**/var/lib/pgsql/data**	postgresql	false
${APPLICATION_NAME}-kieserver-2	kieserver-keystore-volume	**/etc/kieserver-secret-volume**	ssl certs	True
${APPLICATION_NAME}-postgresql-2	**${APPLICATION_NAME}-postgresql-pvol**	**/var/lib/pgsql/data**	postgresql	false

5.1.2.4. External Dependencies

5.1.2.4.1. 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}-postgresql-claim-1	ReadWriteOnce
${APPLICATION_NAME}-postgresql-claim-2	ReadWriteOnce
${APPLICATION_NAME}-smartrouter-claim	ReadWriteMany
${APPLICATION_NAME}-rhpamcentr-claim	ReadWriteMany

5.1.2.4.2. Secrets

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

businesscentral-app-secret smartrouter-app-secret kieserver-app-secret

5.2. OPENSHIFT USAGE QUICK REFERENCE

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

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

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

- To create a project, use the following command:

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

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

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

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

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

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

  ```
  $ oc get pods
  ```

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

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

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

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

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

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

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

  For more information, see Viewing deployment logs.

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

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

  For more information, see Accessing build logs.

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

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

  For more information, see Manual scaling .

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

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

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

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Friday, June 25, 2021.