Red Hat Process Automation Manager 7.2

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

Last Updated: 2020-05-04

Red Hat Process Automation Manager 7.2 Deploying a Red Hat Process Automation Manager 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 7.2 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 managed server environment on Red Hat OpenShift Container Platform to provide an infrastructure to execute services, process applications, and other business assets. You can use Business Central Monitoring to manage and update the processes running on Process Servers in this environment.

Prerequisites

- At least four gigabytes of memory must be available in the OpenShift cluster/namespace.
- The OpenShift project for the deployment must be created.
- You must be logged in to the project using the **oc** command. For more information about the **oc** command-line tool, see the OpenShift CLI Reference. If you want to use the OpenShift Web console to deploy templates, you must also be logged on using the Web console.
- Dynamic persistent volume (PV) provisioning mustbe enabled. Alternatively, if dynamic PV provisioning is not enabled, enough persistent volumes must be available. By default, the following sizes are required:
    - Each deployed replicated set of Process 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.
- If you intend to scale any of the Business Central Monitoring pods, your OpenShift environment supports persistent volumes with ReadWriteMany mode. For information about access mode support in OpenShift Online volume plug-ins, see Access Modes.

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, providing as few or as many containers as necessary for a particular component. You can use standard OpenShift methods to manage the pods and balance the load.

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

- Process Server, also known as *Execution Server* or *KIE 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 Process 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, Process Server can use an H2 database; in this case, the pod cannot be scaled.

  You can freely scale up a Process Server pod, providing as many copies as necessary, running on the same host or different hosts. As you scale a pod up or down, all its copies 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 Process 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 Process Server pods as necessary.

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

IMPORTANT

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

- Business Central Monitoring is a web-based management and monitoring console. It can manage deployment of services to Process 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 Process Servers and other components that interact with them. It is required if you want Business Central or Business Central Monitoring to interact with several different Process Servers. Also, when your environment includes many services running on different Process Servers, Smart Router provides a single endpoint to all client applications. A client application can make a REST API call requiring any service. Smart Router automatically determines which Process Server must be called for any 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 Process 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 Process 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. For instructions about deploying this environment, see *Deploying a Red Hat Process Automation Manager 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 Process 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 Process 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 Process Server or undeploy any existing ones (you can not 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 Process 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 need to complete several preparatory tasks. You do not need to repeat these tasks if you want to deploy additional images, for example, for new versions of 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 of 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 the information about their location (known as *image streams*). 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 the same project.

Procedure

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

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

   ```
   $ oc get imagestreamtag -n openshift | grep rhpam72-businesscentral
   $ oc get imagestreamtag -n openshift | grep rhpam72-kieserver
   ```

   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 was 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 on to Red Hat Customer Portal to access the document and to complete the steps to create a registry service account.

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

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

e. Run the following commands:

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

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

f. Download the **rhpam-7.2.0-openshift-templates.zip** product deliverable file from the Software Downloads page and extract the **rhpam72-image-streams.yaml** file.

g. Complete one of the following actions:

- Run the following command:

```
$ oc create -f rhpam72-image-streams.yaml
```

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

NOTE

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

2.2. CREATING THE SECRETS FOR PROCESS SERVER

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

You must create an SSL certificate for Process 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 Process Server. In a production environment, generate a valid signed certificate that matches the expected URL of the Process Server. Save the keystore in a file named **keystore.jks**. Record the name of the certificate and the password of the keystore file.
   See Generate a SSL Encryption Key and Certificate for more information on how to create a keystore with self-signed or purchased SSL certificates.

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

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

2.3. CREATING THE SECRETS FOR BUSINESS CENTRAL

If you are planning to deploy Business Central or Business Central Monitoring in your OpenShift environment, you must create an SSL certificate for Business Central and provide it to your OpenShift environment as a secret. Do not use the same certificate and keystore for Business Central and for Process Server.

Procedure

1. Generate an SSL keystore with a private and public key for SSL encryption for Business Central. In a production environment, generate a valid signed certificate that matches the expected URL of the Business Central. Save the keystore in a file named **keystore.jks**. Record the name of the certificate and the password of the keystore file.
   See Generate a SSL Encryption Key and Certificate for more information on how to create a keystore with self-signed or purchased SSL certificates.

2. Use the **oc** command to generate a secret named **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

If you are planning to deploy Smart Router in your OpenShift environment, you must create an SSL certificate for 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 Process Server or Business Central.

Procedure

1. Generate an SSL keystore with a private and public key for SSL encryption for Smart Router. In a production environment, generate a valid signed certificate that matches the expected URL of the Smart Router. Save the keystore in a file named **keystore.jks**. Record the name of the certificate and the password of the keystore file.
   See Generate a SSL Encryption Key and Certificate for more information on how to create a keystore with self-signed or purchased SSL certificates.

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

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

2.5. CHANGING GLUSTERFS CONFIGURATION

Check whether your OpenShift environment uses GlusterFS to provide permanent storage volumes. If it uses GlusterFS, to ensure optimal performance, tune your GlusterFS storage by changing the storage class configuration.

Procedure

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

   ```
   oc get storageclass
   ```

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

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

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

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

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

Where **class-name** is the name of the default storage class. For example:

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

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

   a. Remove the lines with the following keys:

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

   b. On the line with the **volumeoptions** key, add the following two options: **features.cache-invalidation on, performance.nl-cache on**. For example:

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

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

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

Where **class-name** is the name of the default storage class. For example:

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

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

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

CHAPTER 3. MANAGED SERVER ENVIRONMENT

You can deploy a managed server environment that includes several different pods running Process Server. By default, no processes are initially loaded on the servers. The database servers are, by default, also run in pods. Each Process 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 Process 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 actually runs the process.

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.

Red Hat Process Automation Manager includes the **rhpam72-prod.yaml** base template a for managed environment. It includes two Process Server pods (with database pods), Smart Router in a high-availability configuration, and Business Central monitoring in a high-availability configuration.

To deploy a managed environment, deploy this template. You can change the number of replicas of all components when configuring the deployment. If you want to modify the number of independent Process Server pods or to use a different database server, you must modify the template. The default database server is PostgreSQL.

3.1. DEPLOYING A MANAGED ENVIRONMENT

Deploy a managed environment using the **rhpam72-prod.yaml** template file. You can extract the file from the **rhpam-7.2.0-openshift-templates.zip** product deliverable file. You can download this file from the Software Downloads page.

If you want to modify the environment defined by the template file, see Section 3.3, “Modifying a template for a managed environment”.

Procedure

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

   - In the OpenShift Web UI, select **Add to Project → Import YAML / JSON**and then select or paste the template 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-file-name> -p
     BUSINESS_CENTRAL_HTTPS_SECRET=businesscentral-app-secret -p
     KIE_SERVER_HTTPS_SECRET=kieserver-app-secret -p
     KIE_SERVER_ROUTER_HTTPS_SECRET=smartrouter-app-secret -p
     MAVEN_REPO_URL=<maven-repo-url>
     ```

     In this command line:

  - Replace **<template-file-name>** with the full pathname of the template file that you have modified.
  - Replace **<maven-repo-url>** with the a URL for a Maven repository where the necessary KJAR files are available.
  - Use as many **-p PARAMETER=value** pairs as needed to set the required parameters. You can view the template file to see descriptions for all parameters.

2. Set the following parameters as necessary:

   - Business Central 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 Process Server, as created in Section 2.2, "Creating the secrets for Process 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".
   - Application Name (**APPLICATION_NAME**): The name of the OpenShift application. It is used in the default URLs for Business Central and Process 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.
   - 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 in your environment into this repository.
   - Maven repository username (**MAVEN_REPO_USERNAME**): The username for the Maven repository.
   - Maven repository password(**MAVEN_REPO_PASSWORD**): The username for the Maven repository.
   - Business Central 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 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" .
   - 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".
   - 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 Process 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 Process Server".
- 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.
- Business Central Monitoring Container Replicas (**BUSINESS_CENTRAL_MONITORING_CONTAINER_REPLICAS**): The number of replicas initially created for Business Central Monitoring. If you do not want to use a high-availability configuration for Business Central Monitoring, set this number to 1.
- Smart Router Container Replicas (**SMART_ROUTER_CONTAINER_REPLICAS**): The number of replicas initially created for Smart Router.
- KIE Server Container Replicas(**KIE_SERVER_CONTAINER_REPLICAS**): The number of replicas initially created for each of the Process Servers.
  You can also set the following user names and passwords:
- KIE Admin User (**KIE_ADMIN_USER**) and KIE Admin Password (**KIE_ADMIN_PWD**): The user name and password for the administrative user in Business Central Monitoring.
- KIE Server User (**KIE_SERVER_USER**) and KIE Server Password (**KIE_SERVER_PWD**): The user name and password that a client application must use to connect to any of the Process Servers.

3. If you want to use RH-SSO or LDAP authentication, complete the following additional configuration. Do not configure LDAP authentication and RH-SSO authentication in the same deployment.

   a. In the RH-SSO or LDAP service, create all user names in the deployment parameters. If you do not set any of the parameters, create users with the default user names. The created users must also be assigned to roles:

      - **KIE_ADMIN_USER**: default user name **adminUser**, roles: **kie-server,rest-all,admin**
      - **KIE_SERVER_CONTROLLER_USER**: default user name **controllerUser**, roles: **kie-server,rest-all,guest**
      - **BUSINESS_CENTRAL_MAVEN_USERNAME** (not needed if you configure the use of an external Maven repository): default user name **mavenUser**. No roles are required.
      - **KIE_SERVER_USER**: default user name **executionUser**, roles **kie-server,rest-all,guest**

   b. If you want to configure Red Hat Single Sign On (RH-SSO) authentication, an RH-SSO realm that applies to Red Hat Process Automation Manager must exist. Process Server. If the client does not yet exist, the template can create it during deployment. Clients within RH-SSO must also exist for Business Central Monitoring and for each Process Server. If the clients do not yet exist, the template can create them during deployment.
      For the user roles that you can configure in RH-SSO, see Roles and users.

      Use one of the following procedures:

i. If the clients for Red Hat Process Automation Manager within RH-SSO already exist, set the following parameters in the template:

   - **RH-SSO URL** (**SSO_URL**): The URL for RH-SSO.
   - **RH-SSO Realm name** (**SSO_REALM**): The RH-SSO realm for Red Hat Process Automation Manager.
   - **Business Central Monitoring RH-SSO Client name** (**BUSINESS_CENTRAL_SSO_CLIENT**): The RH-SSO client name for Business Central Monitoring.
   - **Business Central Monitoring RH-SSO Client Secret** (**BUSINESS_CENTRAL_SSO_SECRET**): The secret string that is set in RH-SSO for the client for Business Central Monitoring.
   - For each Process Server defined in the template:
   - **KIE Server*n* RH-SSO Client name** (**KIE_SERVER*n*_SSO_CLIENT**): The RH-SSO client name for this Process 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 Process Server.
   - **RH-SSO Disable SSL Certificate Validation** (**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**): Set to **true** if your RH-SSO installation does not use a valid HTTPS certificate.

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

   - **RH-SSO URL** (**SSO_URL**): The URL for RH-SSO.
   - **RH-SSO Realm name** (**SSO_REALM**): The RH-SSO realm for Red Hat Process Automation Manager.
   - **Business Central Monitoring RH-SSO Client name** (**BUSINESS_CENTRAL_SSO_CLIENT**): The name of the client to create in RH-SSO for Business Central Monitoring.
   - **Business Central Monitoring RH-SSO Client Secret** (**BUSINESS_CENTRAL_SSO_SECRET**): The secret string to set in RH-SSO for the client for Business Central Monitoring.
   - **Business Central Monitoring Custom http Route Hostname** (**BUSINESS_CENTRAL_HOSTNAME_HTTP**): The fully qualified host name to use for the HTTP endpoint for Business Central Monitoring. If you need to create a client in RH-SSO, you can not leave this parameter blank.
   - **Business Central Monitoring Custom https Route Hostname** (**BUSINESS_CENTRAL_HOSTNAME_HTTPS**): The fully qualified host name to use for the HTTPS endpoint for Business Central Monitoring. If you need to create a client in RH-SSO, you can not leave this parameter blank.
   - For each Process 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 Process 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 Process Server.

    - **KIE Server*n* Custom http Route Hostname** (**KIE_SERVER*n*_HOSTNAME_HTTP**): The fully qualified host name to use for the HTTP endpoint for this Process Server. If you need to create a client in RH-SSO, you can not leave this parameter blank.

    - **KIE Server*n* Custom https Route Hostname** (**KIE_SERVER*n*_HOSTNAME_HTTPS**): The fully qualified host name to use for the HTTPS endpoint for this Process Server. If you need to create a client in RH-SSO, you can not leave this parameter blank.

    - **RH-SSO Realm Admin Username** (**SSO_USERNAME**) and **RH-SSO Realm Admin Password** (**SSO_PASSWORD**): The user name and password for the realm administrator user for the RH-SSO realm for Red Hat 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.

   c. To configure LDAP, set the **AUTH_LDAP*** parameters of the template. These parameters correspond to the settings of the LdatExtended 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 pathname of a file that defines role mapping, for example, **/opt/eap/standalone/configuration/rolemapping/rolemapping.properties**. You must provide this file and mount it at this path in all applicable deployment configurations; for instructions, see Section 3.2, “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**.

4. If you modified the template to use an external database server for the Process Server, as described in Section 3.3, “Modifying a template for a managed environment”, 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_HOST**): The JDBC URL for the external database server.
- **KIE Server External Database Dialect**(**KIE_SERVER_EXTERNALDB_DIALECT**): The Hibernate dialect for the server, depending on the server type:
  - **org.hibernate.dialect.MySQL5Dialect** (used for MySQL and MariaDB)
  - **org.hibernate.dialect.PostgreSQLDialect**
  - **org.hibernate.dialect.SQLServer2012Dialect** (used for MS SQL)
  - **org.hibernate.dialect.DB2Dialect**
  - **org.hibernate.dialect.Oracle12cDialect**
  - **org.hibernate.dialect.SybaseASE15Dialect**
- **KIE Server External Database Host**(**KIE_SERVER_EXTERNALDB_HOST**): The host name of the external database server.
- **KIE Server External Database Port**(**KIE_SERVER_EXTERNALDB_PORT**): The port number of the external database server.
- **KIE Server External Database name**(**KIE_SERVER_EXTERNALDB_DB**): The database name to use on the external database server.

5. If you created a custom image for using an external database server other than MySQL or PostgreSQL, as described in Section 3.4, “Building a custom Process Server image for an external database”, set the KIE Server Image Stream Name (**KIE_SERVER_IMAGE_STREAM_NAME**) parameter to the following value:
   - For Microsoft SQL Server, **rhpam72-kieserver-mssql-openshift**
   - For MariaDB, **rhpam72-kieserver-mariadb-openshift**
   - For IBM DB2, **rhpam72-kieserver-db2-openshift**
   - For Oracle Database, **rhpam72-kieserver-oracle-openshift**
   - For Sybase, **rhpam72-kieserver-sybase-openshift**
6. Complete the creation of the environment. Depending on the method that you are using:
   - 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 and run the command line.

3.2. 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. Run the following command:

   ```
   oc create configmap ldap_role_mapping --from-file=<new_name>=<existing_name>
   ```

   Where **new_name** is 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** is the name of the file that you created. For example:

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

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

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

   Where **myapp** is the application name. Sometimes, several Process 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
   ```

   Where **mapping_dir** is the directory name (without file name) set in the **AUTH_ROLE_MAPPER_ROLES_PROPERTIES** parameter, for example, **/opt/eap/standalone/configuration/rolemapping** .

3.3. MODIFYING A TEMPLATE FOR A MANAGED ENVIRONMENT

To adjust the managed environment to your needs, you need to modify the **rhpam72-sit.yaml** or **rhpam72-prod.yaml** template before deploying the environment.

By default, the templates create two replicated Process Server pods. You can deploy separate processes on each of the pods. To add more replicated Process 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 Process 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 **rhpam72-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 Process 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 **rhpam72-kieserver-postgresql.yaml** file, then modify some of the newly added blocks:

  1. Replace the block named **MySQL database parameters** with the block named **PosgreSQL database parameters**. (Take this block and all subsequent replacement blocks from the **rhpam72-kieserver-postgresql.yaml** file.)

rhpam72-kieserver-postgresql.yaml file.)
Repeat the following actions for every replicated Process 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 **PosgreSQL service N** with the block named **MySQL service**.
- Replace the block named **PosgreSQL driver settings N** with the block named **MySQL driver settings**.
- Replace the block named **PosgreSQL deployment config N** with the block named **MySQL deployment config**.
- Replace the block named **PosgreSQL 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 **rhpam72-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 **rhpam72-kieserver-external.yaml** file.)
  Repeat the following actions for every replicated Process 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 **PosgreSQL service N**
     - Remove the block named **PosgreSQL deployment config N**
     - Remove the block named **PosgreSQL persistent volume claim N**
     - Replace the block named **PosgreSQL 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 Process 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 Process Server image includes drivers for MySQL and PostgreSQL external database servers. If you want to use another database server, you must build a custom Process Server image. For instructions, see Section 3.4, “Building a custom Process Server 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.

3.4. BUILDING A CUSTOM PROCESS SERVER IMAGE FOR AN EXTERNAL DATABASE

If you want to use an external database server for a Process Server and this server is neither MySQL nor PostgreSQL, you must build a custom Process Server image with drivers for this server before deploying your environment.

You can use this build procedure to provide drivers for the following database servers:

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

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

The build procedure creates a custom image that extends the existing Process Server image. It pushes this custom image into a new **ImageStream** in the **openshift** namespace with the same version tag as the original image.

Prerequisites

- You have logged on to your project in the OpenShift environment using the **oc** command as a user with the **cluster-admin** role.
- For IBM DB2, Oracle Database, or Sybase, you have downloaded the JDBC driver from the database server vendor.

Procedure

1. For IBM DB2, Oracle Database, or Sybase, provide the JDBC driver JAR in a local directory or on an HTTP server. Within the local directory or HTTP server, the following paths are expected:
   - For IBM DB2, **<local_path_or_url>/com/ibm/db2/jcc/db2jcc4/10.5/db2jcc4-10.5.jar**
   - For Oracle Database, **<local_path_or_url>/com/oracle/ojdbc7/12.1.0.1/ojdbc7-12.1.0.1.jar**

- For Sybase, **<local_path_or_url>/com/sysbase/jconn4/16.0_PL05/jconn4-16.0_PL05.jar** Where **<local_path_or_url>** is the path to the local directory or the URL for the HTTP server where the driver is provided.

2. To install the source code for the custom build, download the **rhpam-7.2.0-openshift-templates.zip** product deliverable file from the Software Downloads page. Unzip the file and, using the command line, change to the **templates/contrib/jdbc** directory of the unzipped file.

3. Change to the following subdirectory:
   - For Microsoft SQL Server, **mssql-driver-image**
   - For MariaDB, **mariadb-driver-image**
   - For IBM DB2, **db2-driver-image**
   - For Oracle Database, **oracle-driver-image**
   - For Sybase, **sybase-driver-image**

4. Run the following command:
   - For Microsoft SQL Server or MariaDB:

   ```
   ../build.sh
   ```

   - For IBM DB2, Oracle Database, or Sybase:

   ```
   ../build.sh --artifact-repo=<local_path_or_url>
   ```

   Where **<local_path_or_url>** is the path to the local directory or the URL for the HTTP server where the driver is provided. For example:

   ```
   ../build.sh --artifact-repo=/home/builder/drivers
   ../build.sh --artifact-repo=http://nexus.example.com/nexus/content/groups/public
   ```

   If you want to configure your OpenShift docker registry address in the process, add also the **--registry=<registry_name.domain_name:port>** parameter to your build command.

   Examples:

   ```
   ../build.sh --registry=docker-registry.custom-domain:80

   ../build.sh --artifact-repo=/home/builder/drivers --registry=docker-registry.custom-domain:80
   ```

CHAPTER 4. OPENSHIFT TEMPLATE REFERENCE INFORMATION

Red Hat Process Automation Manager provides the following OpenShift templates. To access the templates, download and extract the **rhpam-7.2.0-openshift-templates.zip** product deliverable file from the Software Downloads page of the Red Hat customer portal.

- **rhpam72-prod.yaml** provides a high-availability Business Central Monitoring instance, a Smart Router, two distinct Process Servers connected to the Business Central and to the Smart Router, and two PostgreSQL instances. Each Process 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 4.1, “rhpam72-prod.yaml template”.

4.1. RHPAM72-PROD.YAML TEMPLATE

Application template for a managed HA production runtime environment, for Red Hat Process Automation Manager 7.2

4.1.1. Parameters

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

Variable name	Image Environment Variable	Description	Example value	Required
APPLICATION_NAME	—	The name for the application.	myapp	True
MAVEN_REPO_ID	**MAVEN_REPO_ID**	The id to use for the maven repository, if set. Default is generated randomly.	my-repo-id	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**	Username to access the Maven repository, if required.	—	False

Variable name	Image Environment Variable	Description	Example value	Required
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
BUSINESS_CENTRAL_MAVEN_USERNAME	**RHPAMCENTR_MAVEN_REPO_USERNAME**	Username to access the Maven service hosted by Business Central inside EAP.	mavenUser	False
BUSINESS_CENTRAL_MAVEN_PASSWORD	**RHPAMCENTR_MAVEN_REPO_PASSWORD**	Password to access the Maven service hosted by Business Central inside EAP.	maven1!	False
KIE_ADMIN_USER	**KIE_ADMIN_USER**	KIE administrator username	adminUser	False
KIE_ADMIN_PWD	**KIE_ADMIN_PWD**	KIE administrator password	–	False
KIE_SERVER_USER	**KIE_SERVER_USER**	KIE server username (Sets the org.kie.server.user system property)	executionUser	False
KIE_SERVER_PWD	**KIE_SERVER_PWD**	KIE server password (Sets the org.kie.server.pwd system property)	–	False

Variable name	Image Environment Variable	Description	Example value	Required
IMAGE_STREAM_NAMESPACE	–	Namespace in which the ImageStreams for Red Hat Middleware images are installed. These ImageStreams are normally installed in the openshift namespace. You should only need to modify this if you installed the ImageStreams in a different namespace/project.	openshift	True
KIE_SERVER_IMAGE_STREAM_NAME	–	The name of the image stream to use for KIE server. Default is "rhpam72-kieserver-openshift".	rhpam72-kieserver-openshift	True
IMAGE_STREAM_TAG	–	A named pointer to an image in an image stream. Default is "1.1".	1.1	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_USER	**KIE_SERVER_CONTROLLER_USER**	KIE server controller username (Sets the org.kie.server.controller.user system property)	controllerUser	False
KIE_SERVER_CONTROLLER_PWD	**KIE_SERVER_CONTROLLER_PWD**	KIE server controller password (Sets the org.kie.server.controller.pwd system property)	–	False
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 should only need to modify this if you installed the ImageStream in a different namespace/project. Default is "openshift".	openshift	False

Variable name	Image Environment Variable	Description	Example value	Required
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 username	rhpam	False
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
DROOLS_SERVER_FILTER_CLASSES	**DROOLS_SERVER_FILTER_CLASSES**	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	true	False
KIE_MBEANS	**KIE_MBEANS**	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	enabled	False

Variable name	Image Environment Variable	Description	Example value	Required
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
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

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER1_USE_SECURE_ROUTE_NAME	**KIE_SERVER_USE_SECURE_ROUTE_NAME**	If true, will use secure-APPLICATION_NAME-kieserver-1 vs. APPLICATION_NAME-kieserver-1 as the route name.	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
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	businesscentral-app-secret	True
BUSINESS_CENTRAL_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret	keystore.jks	False

Variable name	Image Environment Variable	Description	Example value	Required
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	smartrouter-app-secret	True
KIE_SERVER_ROUTER_HTTPS_KEYSTORE	–	The name of the keystore file within the secret	keystore.jks	False
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	kieserver-app-secret	True
KIE_SERVER_HTTPS_KEYSTORE	**HTTPS_KEYSTORE**	The name of the keystore file within the secret	keystore.jks	False
KIE_SERVER_HTTPS_NAME	**HTTPS_NAME**	The name associated with the server certificate	jboss	False
KIE_SERVER_HTTPS_PASSWORD	**HTTPS_PASSWORD**	The password for the keystore and certificate	mykeystorepass	False

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_BYPASS_AUTH_USER	**KIE_SERVER_BYPASS_AUTH_USER**	KIE server bypass auth user (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
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, will define how much Business Central Monitoring containers will be started.	3	True
SMART_ROUTER_CONTAINER_REPLICAS	–	Smart Router Container Replicas, will define how much smart router containers will be started.	2	True

Variable name	Image Environment Variable	Description	Example value	Required
KIE_SERVER_CONTAINER_REPLICAS	–	KIE Server Container Replicas, will define how much 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
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 Username used to create the Client if it doesn't exist	–	False

Variable name	Image Environment Variable	Description	Example value	Required
SSO_PASSWORD	**SSO_PASSWORD**	RH-SSO Realm Admin Password used to create the Client	–	False
SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	**SSO_DISABLE_SSL_CERTIFICATE_VALIDATION**	RH-SSO Disable SSL Certificate Validation	false	False
SSO_PRINCIPAL_ATTRIBUTE	**SSO_PRINCIPAL_ATTRIBUTE**	RH-SSO Principal Attribute to use as username.	preferred_username	False
AUTH_LDAP_URL	**AUTH_LDAP_URL**	LDAP Endpoint to connect for authentication	ldap://myldap.example.com	False
AUTH_LDAP_BIND_DN	**AUTH_LDAP_BIND_DN**	Bind DN used for authentication	uid=admin,ou=users,ou=exmample,ou=com	False
AUTH_LDAP_BIND_CREDENTIAL	**AUTH_LDAP_BIND_CREDENTIAL**	LDAP Credentials used for authentication	Password	False
AUTH_LDAP_JAAS_SECURITY_DOMAIN	**AUTH_LDAP_JAAS_SECURITY_DOMAIN**	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	–	False
AUTH_LDAP_BASE_CTX_DN	**AUTH_LDAP_BASE_CTX_DN**	LDAP Base DN of the top-level context to begin the user search.	ou=users,ou=example,ou=com	False

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

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

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

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

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

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

4.1.2. 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.

4.1.2.1. Services

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

Service	Port	Name	Description
${APPLICATION_NAME}-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.

Service	Port	Name	Description
	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.

4.1.2.2. Routes

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

Service	Security	Hostname
${APPLICATION_NAME}-rhpamcentrmon-http	none	**${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}**

Service	Security	Hostname
${APPLICATION_NAME}-kieserver-2-https	TLS passthrough	**${KIE_SERVER2_HOSTNAME_HTTPS}**
${APPLICATION_NAME}-smartrouter-http	none	**${SMART_ROUTER_HOSTNAME_HTTP}**
${APPLICATION_NAME}-smartrouter-https	TLS passthrough	**${SMART_ROUTER_HOSTNAME_HTTPS}**

4.1.2.3. 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. Refer to the Openshift documentation for more information.

4.1.2.3.1. Triggers

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

Deployment	Triggers
${APPLICATION_NAME}-rhpamcentrmon	ImageChange
${APPLICATION_NAME}-smartrouter	ImageChange
${APPLICATION_NAME}-kieserver-1	ImageChange
${APPLICATION_NAME}-postgresql-1	ImageChange
${APPLICATION_NAME}-kieserver-2	ImageChange
${APPLICATION_NAME}-postgresql-2	ImageChange

4.1.2.3.2. 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. Refer to the container-engine documentation for more information.

Deployment	Replicas
${APPLICATION_NAME}-rhpamcentrmon	3
${APPLICATION_NAME}-smartrouter	2

Deployment	Replicas
${APPLICATION_NAME}-kieserver-1	3
${APPLICATION_NAME}-postgresql-1	1
${APPLICATION_NAME}-kieserver-2	3
${APPLICATION_NAME}-postgresql-2	1

4.1.2.3.3. Pod Template

4.1.2.3.3.1. Service Accounts

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

Deployment	Service Account
${APPLICATION_NAME}-kieserver-1	**${APPLICATION_NAME}-kieserver**
${APPLICATION_NAME}-kieserver-2	**${APPLICATION_NAME}-kieserver**

4.1.2.3.3.2. Image

Deployment	Image
${APPLICATION_NAME}-rhpamcentrmon	rhpam72-businesscentral-monitoring-openshift
${APPLICATION_NAME}-smartrouter	rhpam72-smartrouter-openshift
${APPLICATION_NAME}-kieserver-1	**${KIE_SERVER_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-postgresql-1	postgresql
${APPLICATION_NAME}-kieserver-2	**${KIE_SERVER_IMAGE_STREAM_NAME}**
${APPLICATION_NAME}-postgresql-2	postgresql

4.1.2.3.3.3. Readiness Probe

${APPLICATION_NAME}-rhpamcentrmon

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/kie-wb.jsp
```

${APPLICATION_NAME}-kieserver-1

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

${APPLICATION_NAME}-postgresql-1

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

${APPLICATION_NAME}-kieserver-2

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

${APPLICATION_NAME}-postgresql-2

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

4.1.2.3.3.4. Liveness Probe

${APPLICATION_NAME}-rhpamcentrmon

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/kie-wb.jsp
```

${APPLICATION_NAME}-kieserver-1

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

${APPLICATION_NAME}-postgresql-1

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

${APPLICATION_NAME}-kieserver-2

```
/bin/bash -c curl --fail --silent -u '${KIE_ADMIN_USER}:${KIE_ADMIN_PWD}'
http://localhost:8080/services/rest/server/readycheck
```

${APPLICATION_NAME}-postgresql-2

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

4.1.2.3.3.5. 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**

4.1.2.3.3.6. Image Environment Variables

Deployment	Variable name	Description	Example value
${APPLICATION_NAME}-rhpamcentrmon	**KIE_ADMIN_PWD**	KIE administrator password	**${KIE_ADMIN_PWD}**
	KIE_ADMIN_USER	KIE administrator username	**${KIE_ADMIN_USER}**

Deployment	Variable name	Description	Example value
	KIE_SERVER_PWD	KIE server password (Sets the org.kie.server.pwd system property)	**${KIE_SERVER_PWD}**
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	**${KIE_SERVER_USER}**
	MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	**${KIE_SERVER_CONTROLLER_USER}**
	KIE_SERVER_CONTROLLER_PWD	KIE server controller password (Sets the org.kie.server.controller.pwd system property)	**${KIE_SERVER_CONTROLLER_PWD}**
	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**

Deployment	Variable name	Description	Example value
	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	**${BUSINESS_CENTRAL_SSO_SECRET}**
	SSO_CLIENT	Business Central Monitoring RH-SSO Client name	**${BUSINESS_CENTRAL_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn’t exist	**${SSO_USERNAME}**
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	**${SSO_PASSWORD}**

Deployment	Variable name	Description	Example value
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <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	**${AUTH_LDAP_BIND_CREDENTIAL}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	**${AUTH_LDAP_BASE_CTX_DN}**

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

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

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

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

Deployment	Variable name	Description	Example value
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-smartrouter	**KIE_SERVER_ROUTER_HOST**	–	–
	KIE_SERVER_ROUTER_PORT	–	9000
	KIE_SERVER_ROUTER_PORT_TLS	–	9443
	KIE_SERVER_ROUTER_URL_EXTERNAL	Public URL where the router can be found. Format http://<host>:<port> (router property org.kie.server.router.url.external)	**${KIE_SERVER_ROUTER_URL_EXTERNAL}**
	KIE_SERVER_ROUTER_ID	Router ID used when connecting to the controller (router property org.kie.server.router.id)	**${KIE_SERVER_ROUTER_ID}**
	KIE_SERVER_ROUTER_NAME	Router name used when connecting to the controller (router property org.kie.server.router.name)	**${KIE_SERVER_ROUTER_NAME}**

Deployment	Variable name	Description	Example value
	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_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	**${KIE_SERVER_CONTROLLER_USER}**
	KIE_SERVER_CONTROLLER_PWD	KIE server controller password (Sets the org.kie.server.controller.pwd system property)	**${KIE_SERVER_CONTROLLER_PWD}**
	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
${APPLICATION_NAME}-kieserver-1	**DROOLS_SERVER_FILTER_CLASSES**	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	**${DROOLS_SERVER_FILTER_CLASSES}**
	KIE_ADMIN_USER	KIE administrator username	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	KIE administrator password	**${KIE_ADMIN_PWD}**
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE server bypass auth user (Sets the org.kie.server.bypass.auth.user system property)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	**${KIE_SERVER_CONTROLLER_USER}**
	KIE_SERVER_CONTROLLER_PWD	KIE server controller password (Sets the org.kie.server.controller.pwd system property)	**${KIE_SERVER_CONTROLLER_PWD}**
	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

Deployment	Variable name	Description	Example value
	KIE_SERVER_ID	–	**${APPLICATION_NAME}-kieserver-1**
	KIE_SERVER_ROUTE_NAME	–	**${APPLICATION_NAME}-kieserver-1**
	KIE_SERVER_USE_SECURE_ROUTE_NAME	If true, will use secure-APPLICATION_NAME-kieserver-1 vs. APPLICATION_NAME-kieserver-1 as the route name.	**${KIE_SERVER1_USE_SECURE_ROUTE_NAME}**
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	**${KIE_SERVER_USER}**
	KIE_SERVER_PWD	KIE server password (Sets the org.kie.server.pwd system property)	**${KIE_SERVER_PWD}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	–	
	MAVEN_REPOS	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_SERVICE	The service name for the optional Business Central, where it can be reached, to allow service lookups (for maven repo usage), if required	**${BUSINESS_CENTRAL_MAVEN_SERVICE}**
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHPAMCENTR_MAVEN_REPO_USERNAME	Username to access the Maven service hosted by Business Central inside EAP.	**${BUSINESS_CENTRAL_MAVEN_USERNAME}**

Deployment	Variable name	Description	Example value
	RHPAMCENTR_MAVEN_REPO_PASSWORD	Password to access the Maven service hosted by Business Central inside EAP.	**${BUSINESS_CENTRAL_MAVEN_PASSWORD}**
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_ROUTER_SERVICE	–	**${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

Deployment	Variable name	Description	Example value
	RHPAM_DATABASE	KIE server PostgreSQL database name	**${KIE_SERVER_POSTGRESQL_DB}**
	RHPAM_DRIVER	–	postgresql
	KIE_SERVER_PERSISTENCE_DIALECT	–	org.hibernate.dialect.PostgreSQLDialect
	RHPAM_USERNAME	KIE server PostgreSQL database username	**${KIE_SERVER_POSTGRESQL_USER}**
	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**

Deployment	Variable name	Description	Example value
	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 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 Username used to create the Client if it doesn’t exist	**${SSO_USERNAME}**
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${KIE_SERVER1_HOSTNAME_HTTP}**

Deployment	Variable name	Description	Example value
	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}**
	AUTH_LDAP_JAAS_SECURITY_DOMAIN	The JMX ObjectName of the JaasSecurityDomain used to decrypt the password.	**${AUTH_LDAP_JAAS_SECURITY_DOMAIN}**
	AUTH_LDAP_BASE_CTX_DN	LDAP Base DN of the top-level context to begin the user search.	**${AUTH_LDAP_BASE_CTX_DN}**
	AUTH_LDAP_BASE_FILTER	LDAP search filter used to locate the context of the user to authenticate. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. A common example for the search filter is (uid={0}).	**${AUTH_LDAP_BASE_FILTER}**
	AUTH_LDAP_SEARCH_SCOPE	The search scope to use.	**${AUTH_LDAP_SEARCH_SCOPE}**
	AUTH_LDAP_SEARCH_TIME_LIMIT	The timeout in milliseconds for user or role searches.	**${AUTH_LDAP_SEARCH_TIME_LIMIT}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE	The name of the attribute in the user entry that contains the DN of the user. This may be necessary if the DN of the user itself contains special characters, backslash for example, that prevent correct user mapping. If the attribute does not exist, the entry's DN is used.	**${AUTH_LDAP_DISTINGUISHED_NAME_ATTRIBUTE}**
	AUTH_LDAP_PARSE_USERNAME	A flag indicating if the DN is to be parsed for the username. If set to true, the DN is parsed for the username. If set to false the DN is not parsed for the username. This option is used together with usernameBeginString and usernameEndString.	**${AUTH_LDAP_PARSE_USERNAME}**
	AUTH_LDAP_USERNAME_BEGIN_STRING	Defines the String which is to be removed from the start of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_BEGIN_STRING}**
	AUTH_LDAP_USERNAME_END_STRING	Defines the String which is to be removed from the end of the DN to reveal the username. This option is used together with usernameEndString and only taken into account if parseUsername is set to true.	**${AUTH_LDAP_USERNAME_END_STRING}**
	AUTH_LDAP_ROLE_ATTRIBUTE_ID	Name of the attribute containing the user roles.	**${AUTH_LDAP_ROLE_ATTRIBUTE_ID}**

Deployment	Variable name	Description	Example value
	AUTH_LDAP_ROLES_CTX_DN	The fixed DN of the context to search for user roles. This is not the DN where the actual roles are, but the DN where the objects containing the user roles are. For example, in a Microsoft Active Directory server, this is the DN where the user account is.	**${AUTH_LDAP_ROLES_CTX_DN}**
	AUTH_LDAP_ROLE_FILTER	A search filter used to locate the roles associated with the authenticated user. The input username or userDN obtained from the login module callback is substituted into the filter anywhere a {0} expression is used. The authenticated userDN is substituted into the filter anywhere a {1} is used. An example search filter that matches on the input username is (member={0}). An alternative that matches on the authenticated userDN is (member={1}).	**${AUTH_LDAP_ROLE_FILTER}**
	AUTH_LDAP_ROLE_RECURSION	The number of levels of recursion the role search will go below a matching context. Disable recursion by setting this to 0.	**${AUTH_LDAP_ROLE_RECURSION}**
	AUTH_LDAP_DEFAULT_ROLE	A role included for all authenticated users	**${AUTH_LDAP_DEFAULT_ROLE}**

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

Deployment	Variable name	Description	Example value
	AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK	If you are not using referrals, you can ignore this option. When using referrals, this option denotes the attribute name which contains users defined for a certain role, for example member, if the role object is inside the referral. Users are checked against the content of this attribute name. If this option is not set, the check will always fail, so role objects cannot be stored in a referral tree.	**${AUTH_LDAP_REFERRAL_USER_ATTRIBUTE_ID_TO_CHECK}**
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-postgresql-1	**POSTGRESQL_USER**	KIE server PostgreSQL database username	**${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}**

Deployment	Variable name	Description	Example value
	POSTGRESQL_MAX_PREPARED_TRANSACTIONS	Allows the PostgreSQL to handle XA transactions.	**${POSTGRESQL_MAX_PREPARED_TRANSACTIONS}**
${APPLICATION_NAME}-kieserver-2	**DROOLS_SERVER_FILTER_CLASSES**	KIE server class filtering (Sets the org.drools.server.filter.classes system property)	**${DROOLS_SERVER_FILTER_CLASSES}**
	KIE_ADMIN_USER	KIE administrator username	**${KIE_ADMIN_USER}**
	KIE_ADMIN_PWD	KIE administrator password	**${KIE_ADMIN_PWD}**
	KIE_MBEANS	KIE server mbeans enabled/disabled (Sets the kie.mbeans and kie.scanner.mbeans system properties)	**${KIE_MBEANS}**
	KIE_SERVER_BYPASS_AUTH_USER	KIE server bypass auth user (Sets the org.kie.server.bypass.auth.user system property)	**${KIE_SERVER_BYPASS_AUTH_USER}**
	KIE_SERVER_CONTROLLER_USER	KIE server controller username (Sets the org.kie.server.controller.user system property)	**${KIE_SERVER_CONTROLLER_USER}**
	KIE_SERVER_CONTROLLER_PWD	KIE server controller password (Sets the org.kie.server.controller.pwd system property)	**${KIE_SERVER_CONTROLLER_PWD}**
	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

Deployment	Variable name	Description	Example value
	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.	**${KIE_SERVER2_USE_SECURE_ROUTE_NAME}**
	KIE_SERVER_USER	KIE server username (Sets the org.kie.server.user system property)	**${KIE_SERVER_USER}**
	KIE_SERVER_PWD	KIE server password (Sets the org.kie.server.pwd system property)	**${KIE_SERVER_PWD}**
	KIE_SERVER_CONTAINER_DEPLOYMENT	–	
	MAVEN_REPOS	–	RHPAMCENTR,EXTERNAL
	RHPAMCENTR_MAVEN_REPO_SERVICE	The service name for the optional Business Central, where it can be reached, to allow service lookups (for maven repo usage), if required	**${BUSINESS_CENTRAL_MAVEN_SERVICE}**
	RHPAMCENTR_MAVEN_REPO_PATH	–	**/maven2/**
	RHPAMCENTR_MAVEN_REPO_USERNAME	Username to access the Maven service hosted by Business Central inside EAP.	**${BUSINESS_CENTRAL_MAVEN_USERNAME}**
	RHPAMCENTR_MAVEN_REPO_PASSWORD	Password to access the Maven service hosted by Business Central inside EAP.	**${BUSINESS_CENTRAL_MAVEN_PASSWORD}**

Deployment	Variable name	Description	Example value
	EXTERNAL_MAVEN_REPO_ID	The id to use for the maven repository, if set. Default is generated randomly.	**${MAVEN_REPO_ID}**
	EXTERNAL_MAVEN_REPO_URL	Fully qualified URL to a Maven repository or service.	**${MAVEN_REPO_URL}**
	EXTERNAL_MAVEN_REPO_USERNAME	Username to access the Maven repository, if required.	**${MAVEN_REPO_USERNAME}**
	EXTERNAL_MAVEN_REPO_PASSWORD	Password to access the Maven repository, if required.	**${MAVEN_REPO_PASSWORD}**
	KIE_SERVER_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

Deployment	Variable name	Description	Example value
	KIE_SERVER_PERSISTENCE_DIALECT	–	org.hibernate.dialect.PostgreSQLDialect
	RHPAM_USERNAME	KIE server PostgreSQL database username	**${KIE_SERVER_POSTGRESQL_USER}**
	RHPAM_PASSWORD	KIE server PostgreSQL database password	**${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}**
	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}**

Deployment	Variable name	Description	Example value
	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	**${KIE_SERVER2_SSO_CLIENT}**
	SSO_USERNAME	RH-SSO Realm Admin Username used to create the Client if it doesn’t exist	**${SSO_USERNAME}**
	SSO_PASSWORD	RH-SSO Realm Admin Password used to create the Client	**${SSO_PASSWORD}**
	SSO_DISABLE_SSL_CERTIFICATE_VALIDATION	RH-SSO Disable SSL Certificate Validation	**${SSO_DISABLE_SSL_CERTIFICATE_VALIDATION}**
	SSO_PRINCIPAL_ATTRIBUTE	RH-SSO Principal Attribute to use as username.	**${SSO_PRINCIPAL_ATTRIBUTE}**
	HOSTNAME_HTTP	Custom hostname for http service route. Leave blank for default hostname, e.g.: <application-name>-kieserver-<project>.<default-domain-suffix>	**${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}**

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

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

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

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

Deployment	Variable name	Description	Example value
	AUTH_ROLE_MAPPER_ROLES_PROPERTIES	When present, the RoleMapping Login Module will be configured to use the provided file. This property defines the fully-qualified file path and name of a properties file or resource which maps roles to replacement roles. The format is original_role=role1,role2,role3	**${AUTH_ROLE_MAPPER_ROLES_PROPERTIES}**
	AUTH_ROLE_MAPPER_REPLACE_ROLE	Whether to add to the current roles, or replace the current roles with the mapped ones. Replaces if set to true.	**${AUTH_ROLE_MAPPER_REPLACE_ROLE}**
${APPLICATION_NAME}-postgresql-2	**POSTGRESQL_USER**	KIE server PostgreSQL database username	**${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}**

4.1.2.3.3.7. 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

4.1.2.4. External Dependencies

4.1.2.4.1. 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. Refer to 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

4.1.2.4.2. Secrets

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

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

4.2. 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 run 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 run 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 run 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 Tuesday, May 28, 2019.