



Red Hat Process Automation Manager 7.8

Installing and configuring Red Hat Process
Automation Manager in a Red Hat JBoss EAP
clustered environment

Red Hat Process Automation Manager 7.8 Installing and configuring Red Hat Process Automation Manager in a Red Hat JBoss EAP clustered environment

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

Legal Notice

Copyright © 2021 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This document describes how to create a Red Hat Process Automation Manager 7.8 clustered environment on Red Hat JBoss Enterprise Application Platform 7.3.

Table of Contents

PREFACE	3
CHAPTER 1. RED HAT PROCESS AUTOMATION MANAGER CLUSTERS	4
CHAPTER 2. RED HAT PROCESS AUTOMATION MANAGER CLUSTERS IN A DEVELOPMENT (AUTHORING) ENVIRONMENT	5
2.1. INSTALLING AND CONFIGURING RED HAT DATA GRID	5
2.2. DOWNLOADING AND CONFIGURING AMQ BROKER	6
2.3. CONFIGURING AN NFS SERVER	7
2.4. DOWNLOADING AND EXTRACTING RED HAT JBOSS EAP 7.3 AND RED HAT PROCESS AUTOMATION MANAGER	8
2.5. CONFIGURING AND RUNNING BUSINESS CENTRAL IN A CLUSTER	9
2.6. VERIFYING THE RED HAT PROCESS AUTOMATION MANAGER CLUSTER	11
CHAPTER 3. KIE SERVER CLUSTERS IN A RUNTIME ENVIRONMENT	13
3.1. DOWNLOADING AND EXTRACTING RED HAT JBOSS EAP 7.3 AND KIE SERVER	13
3.2. CONFIGURING AND RUNNING A RED HAT JBOSS EAP 7.3 CLUSTER FOR KIE SERVER	14
3.3. CLUSTERING KIE SERVERS WITH THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER	17
CHAPTER 4. INSTALLING AND CONFIGURING SMART ROUTER	19
4.1. LOAD-BALANCING KIE SERVER INSTANCES WITH SMART ROUTER	19
4.2. CONFIGURING SMART ROUTER FOR TLS SUPPORT	22
4.3. CONFIGURING SMART ROUTER FOR ENDPOINT AUTHENTICATION	22
4.4. CONFIGURING SMART ROUTER BEHAVIOR	23
CHAPTER 5. CONFIGURING QUARTZ TIMER SERVICE	25
CHAPTER 6. ADDITIONAL RESOURCES	27
APPENDIX A. VERSIONING INFORMATION	28

PREFACE

As a system engineer, you can create a Red Hat Process Automation Manager clustered environment to provide high availability and load balancing for your development and runtime environments.

Prerequisites

- You have reviewed the information in [Planning a Red Hat Process Automation Manager installation](#).

CHAPTER 1. RED HAT PROCESS AUTOMATION MANAGER CLUSTERS

By clustering two or more computers, you have the benefits of high availability, enhanced collaboration, and load balancing. High availability decreases the chance of a loss of data when a single computer fails. When a computer fails, another computer fills the gap by providing a copy of the data that was on the failed computer. When the failed computer comes online again, it resumes its place in the cluster. Load balancing shares the computing load across the nodes of the cluster. Doing this improves the overall performance.

There are several ways that you can cluster Red Hat Process Automation Manager components. This document describes how to cluster the following scenarios:

- [Chapter 2, *Red Hat Process Automation Manager clusters in a development \(authoring\) environment*](#)
- [Chapter 3, *KIE Server clusters in a runtime environment*](#)

CHAPTER 2. RED HAT PROCESS AUTOMATION MANAGER CLUSTERS IN A DEVELOPMENT (AUTHORING) ENVIRONMENT



NOTE

Configuration of Business Central for high availability is currently technology preview.

Developers can use Red Hat Process Automation Manager to author rules and processes that assist users with decision making.

You can configure Red Hat Process Automation Manager as a clustered development environment to benefit from high availability. With a clustered environment, if a developer is working on **\$node1** and that node fails, that developer's work is preserved and visible on any other node of the cluster.

Most development environments consist of Business Central for creating rules and processes, and at least one KIE Server to test those rules and processes.

To create a Red Hat Process Automation Manager clustered development environment, you must perform the following tasks:

- Configure Red Hat JBoss EAP 7.3 with Red Hat Data Grid 7.3 on a machine.
- Configure AMQ Broker, a Java messaging server (JMS) broker, on a machine.
- Configure an NFS file server on a machine.
- Download Red Hat JBoss EAP 7.3 and Red Hat Process Automation Manager 7.8, then install them on each machine that is to become a cluster node.
- Configure and start Business Central on each cluster node to start the operation of the cluster.

Red Hat Data Grid is built from the Infinispan open-source software project. It is a distributed in-memory key/value data store that has indexing capabilities that enable you to store, search, and analyze high volumes of data quickly and in near-real time. In a Red Hat Process Automation Manager clustered environment, it enables you to perform complex and efficient searches across cluster nodes.

A JMS broker is a software component that receives messages, stores them locally, and forwards the messages to a recipient. AMQ Broker enables your applications to communicate with any messaging provider. It specifies how components such as message-driven beans, Enterprise JavaBeans, and servlets can send or receive messages.

2.1. INSTALLING AND CONFIGURING RED HAT DATA GRID

Install and configure Red Hat Data Grid for the Red Hat Process Automation Manager clustered environment for more efficient searching across cluster nodes.

Use the following instructions to configure a simplified, non-high availability environment on a separate machine.

For information about advanced installation and configuration options, and Red Hat Data Grid modules for Red Hat JBoss EAP, see the [Red Hat Data Grid User Guide](#).

**NOTE**

Do not install Red Hat Data Grid on the same node as Business Central.

Prerequisites

- A Java Virtual Machine (JVM) environment compatible with Java 8.0 or later is installed.
- A backed-up Red Hat JBoss EAP installation version 7.3 or higher is available. The base directory of the Red Hat JBoss EAP installation is referred to as **EAP_HOME**.
- Red Hat Process Automation Manager is installed and configured.
- Sufficient user permissions to complete the installation are granted.

Procedure

1. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - **Product:** Data Grid
 - **Version:** 7.3
2. Download and unzip the **Red Hat JBoss Data Grid 7.3.0 Server(jboss-datagrid-7.3.0-1-server.zip)** installation file to the preferred location on your system. The unzipped directory is referred to as **JDG_HOME**.
3. To run Red Hat Data Grid, navigate to **JDG_HOME/bin** and enter one of the following commands:

- On Linux or UNIX-based systems:

```
┆ $ ./standalone.sh -c clustered.xml
```

- On Windows:

```
┆ standalone.bat -c clustered.xml
```

**NOTE**

Updating Red Hat Data Grid to the latest version is recommended. For more information, see Red Hat Data Grid [Red Hat Data Grid User Guide](#).

2.2. DOWNLOADING AND CONFIGURING AMQ BROKER

Red Hat AMQ Broker enables your applications to communicate with any messaging provider. It specifies how components such as message-driven beans, Enterprise JavaBeans, and servlets can send or receive messages.

To configure AMQ Broker for a high availability Red Hat Process Automation Manager clustered environment, see [Getting started with AMQ Broker](#).

You can use the following procedure to configure a simplified, non-high availability environment.

Procedure

1. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - **Product: AMQ Broker**
 - **Version: 7.5.0**
2. Click **Download** next to **Red Hat AMQ Broker 7.5.0(amq-broker-7.5.0-bin.zip)**.
3. Extract the **amq-broker-7.5.0-bin.zip** file.
4. Change directory to **amq-broker-7.5.0-bin/amq-broker-7.5.0/bin**.
5. Enter the following command and replace the following placeholders to create the broker and broker user:
 - **<HOST>** is the IP address or host name of the server where you installed AMQ Broker.
 - **<AMQ_USER>** and **<AMQ_PASSWORD>** is a user name and password combination of your choice.
 - **<BROKER_NAME>** is a name for the broker that you are creating.

```
./artemis create --host <HOST> --user <AMQ_USER> --password <AMQ_PASSWORD>
--require-login <BROKER_NAME>
```

6. To run AMQ Broker, enter the following command in the **amq-broker-7.5.0-bin/amq-broker-7.5.0/bin** directory:

```
amq-broker-7.5.0/bin/<BROKER_NAME>/bin/artemis run
```

2.3. CONFIGURING AN NFS SERVER

A shared file system is required for a Business Central clustered environment and each client node must have access to the shared file system.

You must deploy and configure an NFS version 4 server.

Procedure

1. Configure a server to export NFS version 4 shares. For instructions about exporting NFS shares on Red Hat Enterprise Linux, see *Exporting NFS shares* in [Managing file systems](#). For more information about creating the NFS server, see [How to configure NFS in RHEL 7](#).
2. On the server, create an **/opt/kie/data** share with the **rw,sync,no_root_squash** options by adding the following line to the **/etc/exports** file:

```
/opt/kie/data *(rw,sync,no_root_squash)
```

In this example, **/opt/kie/data** is the shared folder, ***** are the IP addresses allowed to connect to the NFS server, and **(rw,sync,no_root_squash)** are the minimum options required for NFS. For example:

```
/opt/kie/data 192.268.1.0/24(rw,sync,no_root_squash)
```



NOTE

You can use another share name instead of '/opt/kie/data'. In this case, you must use this name when configuring all nodes that run Business Central.

- On each client node, mount the shared folder in an existing directory:

```
# mount <SERVER_IP>:/opt/kie/data /opt/kie/data/niogit
```

- Add the following properties to the **standalone-full-ha.xml** file to bind the **.niogit** and **maven-repository** directories as nfs shared folders:

```
<property name="org.uberfire.nio.git.dir" value="/opt/kie/data/niogit"/>
<property name="org.guvnor.m2repo.dir" value="/opt/kie/data/maven-repository"/>
```

2.4. DOWNLOADING AND EXTRACTING RED HAT JBOSS EAP 7.3 AND RED HAT PROCESS AUTOMATION MANAGER

Download and install Red Hat JBoss EAP 7.3 and Red Hat Process Automation Manager 7.8 on each node of the cluster.

Procedure

- Download Red Hat JBoss EAP 7.3 on each node of the cluster:
 - Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - Product: Enterprise Application Platform**
 - Version: 7.3**
 - Click **Download** next to **Red Hat JBoss Enterprise Application Platform 7.3.0 (JBEAP-7.3.0/jboss-eap-7.3.0.zip)**.
- Extract the **jboss-eap-7.3.0.zip** file. In the following steps, **EAP_HOME** is the **jboss-eap-7.3/jboss-eap-7.3** directory.
- Download and apply the latest Red Hat JBoss EAP patch, if available.
- Download Red Hat Process Automation Manager on each node of the cluster:
 - Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal, and select the product and version from the drop-down options:
 - Product: Process Automation Manager**
 - Version: 7.8**
 - Download **Red Hat Process Automation Manager 7.8.0 Business Central Deployable for Red Hat JBoss EAP 7 (rhpam-7.8.0-business-central-eap7-deployable.zip)**.

5. Extract the **rhpm-7.8.0-business-central-eap7-deployable.zip** file to a temporary directory. In the following commands this directory is called **TEMP_DIR**.
6. Copy the contents of **TEMP_DIR/rhpm-7.8.0-business-central-eap7-deployable/jboss-eap-7.3** to **EAP_HOME**.
7. Download and apply the latest Red Hat Process Automation Manager patch, if available.
8. Navigate to the **EAP_HOME/bin** directory.
9. Create a user with the **admin** role that you will use to log in to Business Central. In the following command, replace **<username>** and **<password>** with the user name and password of your choice.

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role admin,rest-all
```



NOTE

Make sure that the specified user name is not the same as an existing user, role, or group. For example, do not create a user with the user name **admin**.

The password must have at least eight characters and must contain at least one number and one non-alphanumeric character, but not & (ampersand).

You must use LDAP or RH-SSO for high availability environments. For more information, see the [Red Hat Single Sign-On Server Administration Guide](#) .

10. Create a user with the **kie-server** role that you will use to log in to KIE Server.

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role kie-server
```

11. Make a note of your user names and passwords.

2.5. CONFIGURING AND RUNNING BUSINESS CENTRAL IN A CLUSTER

After you install Red Hat JBoss EAP and Business Central you can use Red Hat Data Grid and the AMQ Broker to configure the cluster. Complete these steps on each node of the cluster.



NOTE

These steps describe a basic cluster configuration. For more complex configurations, see the [Red Hat JBoss EAP 7.3 Configuration Guide](#) .

Prerequisites

- Red Hat Data Grid 7.3 is installed as described in [Section 2.1, “Installing and configuring Red Hat Data Grid”](#) .
- AMQ Broker is installed and configured, as described in [Section 2.2, “Downloading and configuring AMQ Broker”](#) .
- Red Hat JBoss EAP and Red Hat Process Automation Manager are installed on each node of the cluster as described in [Section 2.4, “Downloading and extracting Red Hat JBoss EAP 7.3 and Red Hat Process Automation Manager”](#) .

- An NFS server with a mounted partition is available as described in [Section 2.3, "Configuring an NFS server"](#).

Procedure

1. Mount the directory shared over NFS as **/data**. Enter the following commands as the root user:

```
mkdir /data
mount <NFS_SERVER_IP>:<DATA_SHARE> /data
```

Replace `<NFS_SERVER_IP>` with the IP address or hostname of the NFS server machine.
Replace `<DATA_SHARE>` with the share name that you configured (for example, `/opt/kie/data`).

2. Open the **`EAP_HOME/standalone/configuration/standalone-full.xml`** file in a text editor.
3. Edit or add the properties under the **`<system-properties>`** element and replace the following placeholders:

- **`<AMQ_USER>`** and **`<AMQ_PASSWORD>`** are the credentials that you defined when creating the AMQ Broker.
- **`<AMQ_BROKER_IP_ADDRESS>`** is the IP address of the AMQ Broker.
- **`<INFINISPAN_NODE_IP>`** is the IP address where Red Hat Data Grid is installed.

```
<system-properties>
  <property name="appformer-jms-connection-mode" value="REMOTE"/>
  <property name="appformer-jms-username" value="<AMQ_USER>"/>
  <property name="appformer-jms-password" value="<AMQ_USER_PASSWORD>"/>
  <property name="appformer-jms-url"
    value="tcp://<AMQ_BROKER_IP_ADDRESS>:61616?
ha=true&retryInterval=1000&retryIntervalMultiplier=1.0&reconnectAttempts=
-1"/>
  <property name="org.appformer.ext.metadata.infinispan.port"
    value="11222"/>
  <property name="org.appformer.ext.metadata.infinispan.host"
    value="<INFINISPAN_NODE_IP>"/>
  <property name="org.appformer.ext.metadata.infinispan.realm"
    value="ApplicationRealm"/>
  <property name="org.appformer.ext.metadata.infinispan.cluster"
    value="kie-cluster"/>
  <property name="org.appformer.ext.metadata.index"
    value="infinispan"/>
  <property name="org.uberfire.nio.git.dir"
    value="/data"/>
  <property name="es.set.netty.runtime.available.processors"
    value="false"/>
</system-properties>
```

4. Optional: If the Red Hat Data Grid deployment requires authentication, edit or add the properties under the **`<system-properties>`** element and replace the following placeholders:

- **`<SERVER_NAME>`** is the server name specified in your Red Hat Data Grid server configuration.

- **<SASL_QOP>** is the combination of auth, auth-int and auth-conf values for your Red Hat Data Grid server configuration.

```
<property name="org.appformer.ext.metadata.infinispan.server.name"
  value="<SERVER_NAME>"/>
<property name="org.appformer.ext.metadata.infinispan.sasl.qop"
  value="<SASL_QOP>"/>
<property name="org.appformer.ext.metadata.infinispan.username"
  value=""/>
<property name="org.appformer.ext.metadata.infinispan.password"
  value=""/>
```

5. Save the **standalone-full.xml** file.
6. To start the cluster, navigate to **EAP_HOME/bin** and enter one of the following commands:
 - On Linux or UNIX-based systems:

```
$ ./standalone.sh -c standalone-full.xml
```

- On Windows:

```
standalone.bat -c standalone-full.xml
```

2.6. VERIFYING THE RED HAT PROCESS AUTOMATION MANAGER CLUSTER

After configuring the cluster for Red Hat Process Automation Manager, create an asset to verify that the installation is working.

Procedure

1. In a web browser, enter **<node-IP-address>:8080/business-central**. Replace **<node-IP-address>** with the IP address of a particular node.
2. Enter the **admin** user credentials that you created during installation. The Business Central home page appears.
3. Select **Menu → Design → Projects**.
4. Open the **MySpace** space.
5. Click **Try Samples → Mortgages → OK**. The **Assets** window appears.
6. Click **Add Asset → Data Object**.
7. Enter **MyDataObject** in the **Data Object** field and click **OK**.
8. Click **Spaces → MySpace → Mortgages** and confirm that **MyDataObject** is in the list of assets.
9. Enter the following URL in a web browser, where **<node_IP_address>** is the address of a different node of the cluster:
http://<node_IP_address>:8080/business-central

10. Enter the same credentials that you used to log in to Business Central on the first node, where you created the **MyDataObject** asset.
11. Select **Menu** → **Design** → **Projects**.
12. Open the **MySpace** space.
13. Select the **Mortgages** project.
14. Verify that **MyDataObject** is in the asset list.
15. Delete the **Mortgages** project.

CHAPTER 3. KIE SERVER CLUSTERS IN A RUNTIME ENVIRONMENT

In a runtime environment, KIE Server runs services that contain rules and processes that support business decisions. The primary benefit of clustering a KIE Server runtime environment is load balancing. If activity on one node of the cluster increases, that activity can be shared among the other nodes of the cluster to improve performance.

To create a KIE Server clustered runtime environment, you download and extract Red Hat JBoss EAP 7.3 and KIE Server. Then, you configure Red Hat JBoss EAP 7.3 for a cluster, start the cluster, and install KIE Server on each cluster node.

Optionally, you can then cluster the headless Process Automation Manager controller and Smart Router.

3.1. DOWNLOADING AND EXTRACTING RED HAT JBOSS EAP 7.3 AND KIE SERVER

Complete the steps in this section to download and install Red Hat JBoss EAP 7.3 and KIE Server for installation in a clustered environment.

Procedure

1. Download Red Hat JBoss EAP 7.3 on each node of the cluster:
 - a. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required) and select the product and version from the drop-down options:
 - **Product: Red Hat JBoss EAP**
 - **Version: 7.3**
 - b. Click **Download** next to **Red Hat JBoss Enterprise Application Platform 7.3.0 (jboss-eap-7.3.0.zip)**.
2. Extract the **jboss-eap-7.3.0.zip** file. The **jboss-eap-7.3/jboss-eap-7.3** directory is referred to as **EAP_HOME**.
3. Download and apply the latest Red Hat JBoss EAP patch, if available.
4. Download KIE Server:
 - a. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal and select the product and version from the drop-down options:
 - **Product: Process Automation Manager**
 - **Version: 7.8**
 - b. Download **Red Hat Process Automation Manager 7.8.0 KIE Server for All Supported EE8 Containers (rhpam-7.8.0-kie-server-ee8.zip)**.
5. Extract the **rhpam-7.8.0-kie-server-ee8.zip** archive to a temporary directory. In the following examples this directory is called **TEMP_DIR**.
6. Copy the **TEMP_DIR/rhpam-7.8.0-kie-server-ee8/rhpam-7.8.0-kie-server-ee8/kie-server.war** directory to **EAP_HOME/standalone/deployments/**.

**WARNING**

Ensure the names of the Red Hat Process Automation Manager deployments you copy do not conflict with your existing deployments in the Red Hat JBoss EAP instance.

7. Copy the contents of the **TEMP_DIR/rhpam-7.8.0-kie-server-ee8/rhpam-7.8.0-kie-server-ee8/SecurityPolicy/** to **EAP_HOME/bin**. When asked to overwrite files, click **Replace**.
8. In the **EAP_HOME/standalone/deployments/** directory, create an empty file named **kie-server.war.dodeploy**. This file ensures that KIE Server is automatically deployed when the server starts.
9. Download and apply the latest Red Hat Process Automation Manager patch, if available.
10. Navigate to the **EAP_HOME/bin** directory.
11. Create a user with the **kie-server** role that you will use to log in to KIE Server.

```
$ ./add-user.sh -a --user <USERNAME> --password <PASSWORD> --role kie-server
```

12. Make a note of your user names and passwords.

3.2. CONFIGURING AND RUNNING A RED HAT JBOSS EAP 7.3 CLUSTER FOR KIE SERVER

Configure the Red Hat JBoss EAP cluster for KIE Server, and then start the cluster.

Procedure

1. Install the JDBC driver on all Red Hat JBoss EAP instances that are part of this cluster. For more information, see the "[JDBC Drivers](#)" section of the *Red Hat JBoss EAP 7.3 Configuration Guide*.
2. Open the **EAP_HOME/standalone/configuration/standalone-full.xml** file in a text editor.
3. Edit the **data-stores** property and the **timer-service thread-pool-name** above it:
 - The **datasource-jndi-name** is the JNDI name of the database specified at the beginning of this procedure.
 - You can enter any name for the value of the **partition** property. However, a node will only see timers from other nodes that have the same partition name. Grouping nodes into partitions by assigning partition names enables you to break a large cluster up into several smaller clusters. Doing this improves performance. For example, instead of having a cluster of 100 nodes, where all 100 nodes are trying to execute and refresh the same timers, you can create 20 clusters of 5 nodes by giving every group of 5 a different partition name.
 - Replace the **default-data-store** attribute value with **ejb_timer_ds**.

- Set the value of **refresh-interval** in milliseconds to specify how often the EJB timer connects to the database to synchronize and load tasks to be processed.

```
<timer-service thread-pool-name="default" default-data-store="ejb_timer_ds">
<data-stores>
  <database-data-store name="ejb_timer_ds" datasource-jndi-
name="java:jboss/datasources/ejb_timer" database="postgresql"
partition="ejb_timer_part" refresh-interval="30000"/>
</data-stores>
</timer-service>
```

The following table lists the supported databases and the corresponding **database** attribute value:

Table 3.1. Supported databases

Database	Attribute value
Hyper SQL (for demonstration purposes only, not supported)	hsql
PostgreSQL	postgresql
Oracle	oracle
IBM DB2	db2
Microsoft SQL Server	mssql
MySQL and MariaDB	mysql

4. Add the KIE Server and EJB timer data sources to the **standalone-full.xml** file. In these examples, **<DATABASE>** is the name of the database, **<SERVER_NAME>** is the host name of the JNDI database, and **<USER_NAME>** and **<USER_PWD>** are the credentials for that database.

- Add the data source to allow KIE Server to connect to the database, for example:

```
<xa-datasource jndi-name="java:/jboss/datasources/rhpam" pool-name="rhpam-RHPAM"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>
```

- Add the data source to enable the EJB timer, for example:

```

<xa-datasource jndi-name="java:jboss/datasources/ejb_timer" pool-name="ejb_timer"
use-java-context="true" enabled="true">
  <xa-datasource-property name="DatabaseName"><DATABASE></xa-datasource-
property>
  <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
  <xa-datasource-property name="ServerName"><SERVER_NAME></xa-datasource-
property>
  <driver>postgresql</driver>
  <transaction-isolation>TRANSACTION_READ_COMMITTED</transaction-isolation>
  <security>
    <user-name><USER_NAME></user-name>
    <password><USER_PWD></password>
  </security>
</xa-datasource>

```



WARNING

You must use two different databases for KIE Server runtime data and EJB timer data.

5. Add the following properties to the **<system-properties>** element and replace the following placeholders:

- **<JNDI_NAME>** is the JNDI name of your data source. For Red Hat Process Automation Manager, this is **java:/jboss/datasources/rhpam**.
- **<DIALECT>** is the hibernate dialect for your database. The following dialects are supported:
 - DB2: **org.hibernate.dialect.DB2Dialect**
 - MSSQL: **org.hibernate.dialect.SQLServer2012Dialect**
 - MySQL: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - MariaDB: **org.hibernate.dialect.MySQL5InnoDBDialect**
 - Oracle: **org.hibernate.dialect.Oracle10gDialect**
 - PostgreSQL: **org.hibernate.dialect.PostgreSQL82Dialect**
 - PostgreSQL plus: **org.hibernate.dialect.PostgresPlusDialect**

```

<system-properties>
  <property name="org.kie.server.persistence.ds" value="<JNDI_NAME>"/>
  <property name="org.kie.server.persistence.dialect" value="<DIALECT>"/>
  <property name="org.jbpm.ejb.timer.tx" value="true"/>
</system-properties>

```

6. Save the **standalone-full.xml** file.

7. To start the cluster, navigate to **EAP_HOME/bin** and enter one of the following commands:

- On Linux or UNIX-based systems:

```
┌ $ ./standalone.sh -c standalone-full.xml
```

- On Windows:

```
┌ standalone.bat -c standalone-full.xml
```

3.3. CLUSTERING KIE SERVERS WITH THE HEADLESS PROCESS AUTOMATION MANAGER CONTROLLER

The Process Automation Manager controller is integrated with Business Central. However, if you do not install Business Central, you can install the headless Process Automation Manager controller and use the REST API or the KIE Server Java Client API to interact with it.

Prerequisites

- A backed-up Red Hat JBoss EAP installation version 7.3 or later is available. The base directory of the Red Hat JBoss EAP installation is referred to as **EAP_HOME**.
- Sufficient user permissions to complete the installation are granted.
- An NFS server with a mounted partition is available as described in [Section 2.3, “Configuring an NFS server”](#).

Procedure

1. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - **Product: Process Automation Manager**
 - **Version: 7.8**
2. Download **Red Hat Process Automation Manager 7.8.0 Add Ons**(the **rhpm-7.8.0-add-ons.zip** file).
3. Unzip the **rhpm-7.8.0-add-ons.zip** file. The **rhpm-7.8.0-controller-ee7.zip** file is in the unzipped directory.
4. Extract the **rhpm-7.8.0-controller-ee7** archive to a temporary directory. In the following examples this directory is called **TEMP_DIR**.
5. Copy the **TEMP_DIR/rhpm-7.8.0-controller-ee7/controller.war** directory to **EAP_HOME/standalone/deployments/**.

**WARNING**

Ensure that the names of the headless Process Automation Manager controller deployments you copy do not conflict with your existing deployments in the Red Hat JBoss EAP instance.

6. Copy the contents of the **`TEMP_DIR/rhpam-7.8.0-controller-ee7/SecurityPolicy/`** directory to **`EAP_HOME/bin`**. When asked to overwrite files, select **Yes**.
7. In the **`EAP_HOME/standalone/deployments/`** directory, create an empty file named **`controller.war.dodeploy`**. This file ensures that the headless Process Automation Manager controller is automatically deployed when the server starts.
8. Open the **`EAP_HOME/standalone/configuration/standalone.xml`** file in a text editor.
9. Add the following properties to the **`<system-properties>`** element and replace **`<NFS_STORAGE>`** with the absolute path to the NFS storage where the template configuration is stored:

```
<system-properties>
  <property name="org.kie.server.controller.templatefile.watcher.enabled" value="true"/>
  <property name="org.kie.server.controller.templatefile" value="<NFS_STORAGE>"/>
</system-properties>
```

Template files contain default configurations for specific deployment scenarios.

If the value of the **`org.kie.server.controller.templatefile.watcher.enabled`** property is set to true, a separate thread is started to watch for modifications of the template file. The default interval for these checks is 30000 milliseconds and can be further controlled by the **`org.kie.server.controller.templatefile.watcher.interval`** system property. If the value of this property is set to false, changes to the template file are detected only when the server restarts.

10. To start the headless Process Automation Manager controller, navigate to **`EAP_HOME/bin`** and enter the following command:

- On Linux or UNIX-based systems:

```
$ ./standalone.sh
```

- On Windows:

```
standalone.bat
```

CHAPTER 4. INSTALLING AND CONFIGURING SMART ROUTER

Smart Router (KIE Server router) is a lightweight Java component that you can use as an integration layer between multiple KIE Servers, client applications, and other components. Depending on your deployment and execution environment, Smart Router can aggregate multiple independent KIE Server instances as though they are a single server. Smart Router provides the following features:

Data aggregation

Collects data from all KIE Server instances (one instance from each group) when there is a client application request and aggregates the results in a single response.

Routing

Functions as a single endpoint that receives calls from client applications to any of your services and routes each call automatically to the KIE Server that runs the specific service. This means that KIE Servers do not need to have the same services deployed.

Load balancing

Provides efficient load balancing. Load balancing requests for a Smart Router cluster must be managed externally with standard load balancing tools.

Authentication

Authenticates KIE Server instances by using a system property flag and can enable HTTPS traffic.

Environment Management

Manages the changing environment, for example adding or removing server instances.

4.1. LOAD-BALANCING KIE SERVER INSTANCES WITH SMART ROUTER

You can use Smart Router to aggregate multiple independent KIE Server instances as though they are a single server. It performs the role of an intelligent load balancer because it can route requests to individual KIE Server instances and aggregate data from different KIE Server instances. Smart Router uses aliases to perform as a proxy.

Prerequisites

- Multiple KIE Server instances are installed.



NOTE

You do not need to configure KIE Server as unmanaged for Smart Router.

An unmanaged KIE Server does not connect to the controller.

For example, if you connect an unmanaged KIE Server to Smart Router and register Smart Router with the controller, then Business Central contacts the unmanaged KIE Server by using Smart Router.

Procedure

1. Navigate to the [Software Downloads](#) page in the Red Hat Customer Portal (login required), and select the product and version from the drop-down options:
 - **Product: Process Automation Manager**

- **Version: 7.8**
2. Download **Red Hat Process Automation Manager 7.8.0 Add-Ons**
 3. Extract the downloaded **rhpam-7.8.0-add-ons.zip** file to a temporary directory. The **rhpam-7.8.0-smart-router.jar** file is in the extracted **rhpam-7.8.0-add-ons** directory.
 4. Copy the **rhpam-7.8.0-smart-router.jar** file to the location where you will run the file.
 5. Enter the following command to start Smart Router:

```
java
-Dorg.kie.server.router.host=<ROUTER_HOST>
-Dorg.kie.server.router.port=<ROUTER_PORT>
-Dorg.kie.server.controller=<CONTROLLER_URL>
-Dorg.kie.server.controller.user=<CONTROLLER_USER>
-Dorg.kie.server.controller.pwd=<CONTROLLER_PWD>
-Dorg.kie.server.router.config.watcher.enabled=true
-Dorg.kie.server.router.repo=<NFS_STORAGE>
-jar rhpam-7.8.0-smart-router.jar
```

The properties in the preceding command have the following default values:

```
org.kie.server.router.host=localhost
org.kie.server.router.port=9000
org.kie.server.controller= N/A
org.kie.server.controller.user=kieserver
org.kie.server.controller.pwd=kieserver1!
org.kie.server.router.repo= <CURRENT_WORKING_DIR>
org.kie.server.router.config.watcher.enabled=false
```

org.kie.server.controller is the URL of the server controller, for example:

```
org.kie.server.controller=http://<HOST>:<PORT>/controller/rest/controller
```

org.kie.server.router.config.watcher.enabled is an optional settings to enable the watcher service system property.

6. On every KIE Server instance that must connect to the Smart Router, set the **org.kie.server.router** system property to the Smart Router URL.
7. To access Smart Router from the client side, use the Smart Router URL instead of the KIE Server URL, for example:

```
KieServicesConfiguration config =
KieServicesFactory.newRestConfiguration("http://smartrouter.example.com:9000",
"USERNAME", "PASSWORD");
```

In this example, **smartrouter.example.com** is the Smart Router URL, and **USERNAME** and **PASSWORD** are the log in credentials for the Smart Router configuration.



NOTE

You must create containers directly against the **kie-server**. For example:




```
$ curl -v -X POST -H 'Content-type: application/xml' -H 'X-KIE-Content-Type: xstream'
-d @create-container.xml -u ${KIE_CRED} http://${KIE_SERVER_HOST}:${KIE_SERVER_PORT}/kie-server/services/rest/server/config/
```

```
$ cat create-container.xml
<script>
  <create-container>
    <container container-id="example:timer-test:1.1">
      <release-id>
        <group-id>example</group-id>
        <artifact-id>timer-test</artifact-id>
        <version>1.1</version>
      </release-id>
      <config-items>
        <itemName>RuntimeStrategy</itemName>
        <itemValue>PER_PROCESS_INSTANCE</itemValue>
        <itemType></itemType>
      </config-items>
    </container>
  </create-container>
</script>
```

A message about the deployed container is displayed in the smart-router console. For example:

```
INFO: Added http://localhost:8180/kie-server/services/rest/server as server location for
container example:timer-test:1.1
```

To display a list of containers, enter the following command:

```
$ curl http://localhost:9000/mgmt/list
```

The list of containers is displayed:

```
{
  "containerInfo": [{
    "alias": "timer-test",
    "containerId": "example:timer-test:1.1",
    "releaseId": "example:timer-test:1.1"
  }],
  "containers": [
    {"example:timer-test:1.1": ["http://localhost:8180/kie-server/services/rest/server"]},
    {"timer-test": ["http://localhost:8180/kie-server/services/rest/server"]}
  ],
  "servers": [
    {"kieserver2": []},
    {"kieserver1": ["http://localhost:8180/kie-server/services/rest/server"]}
  ]
}
```

To initiate a process using the Smart Router URL, enter the following command:

```
$ curl -s -X POST -H 'Content-type: application/json' -H 'X-KIE-Content-Type: json' -d
'{"timerDuration": "9s"}' -u kieserver:kieserver1!
```



http://localhost:9000/containers/example:timer-test:1.1/processes/timer-test.TimerProcess/instances

4.2. CONFIGURING SMART ROUTER FOR TLS SUPPORT

You can configure Smart Router (KIE Server Router) for TLS support to allow HTTPS traffic.

Prerequisites

- KIE Server is installed on each node of a Red Hat JBoss EAP 7.3 cluster.
- Smart Router is installed and configured. For more information, see [Section 4.1, “Load-balancing KIE Server instances with Smart Router”](#).

Procedure

- To start Smart Router with TLS support and HTTPS enabled, use the TLS keystore properties, for example:

```
java -Dorg.kie.server.router.tls.keystore = <KEYSTORE_PATH>
-Dorg.kie.server.router.tls.keystore.password = <KEYSTORE_PWD>
-Dorg.kie.server.router.tls.keystore.keyalias = <KEYSTORE_ALIAS>
-Dorg.kie.server.router.tls.port = <HTTPS_PORT>
-jar rhpam-7.8.0-smart-router.jar
```

org.kie.server.router.tls.port is a property used to configure the HTTPS port. The default HTTPS port value is **9443**.

4.3. CONFIGURING SMART ROUTER FOR ENDPOINT AUTHENTICATION

You can configure Smart Router (KIE Server Router) for endpoint authentication.

Prerequisites

- KIE Server is installed on each node of a Red Hat JBoss EAP 7.3 cluster.
- Smart Router is installed and configured. For more information, see [Section 4.1, “Load-balancing KIE Server instances with Smart Router”](#).

Procedure

- To start Smart Router with endpoint authentication enabled, configure the management credentials:
 - a. Add the following properties to your KIE Server configuration:

```
`org.kie.server.router.management.username`
`org.kie.server.router.management.password`
```

The default **username** is the KIE Server ID.

- b. Add the following property to your Smart Router configuration:

-

```
`org.kie.server.router.management.password`
```

The **password** property values are **true** or **false** (default).



NOTE

Enabling endpoint authentication means any any operation that lists, adds or removes containers must be authenticated.

1. Optionally, you can add users to Smart Router. For example:

```
java -jar rhpam-7.8.0-smart-router.jar -addUser <USERNAME> <PASSWORD>
```

2. Optionally, you can remove users from Smart Router. For example:

```
java -jar rhpam-7.8.0-smart-router.jar -removeUser <USERNAME>
```

4.4. CONFIGURING SMART ROUTER BEHAVIOR

In a clustered environment with multiple KIE Servers, the default behavior is to send requests to each KIE Server in parallel and a host of each KIE Server is sent the request using the "round-robin" method. In the following example environment, each KIE Server is deployed with the same KJAR but each KJAR version is different:

Table 4.1. Example environment

Server Name	KJAR version	Hosts
kie-server1	kjar:1.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=1.0)	129.0.1.1, 129.0.1.2, 129.0.1.3
kie-server2	kjar:2.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=2.0)	129.0.2.1, 129.0.2.2, 129.0.2.3
kie-server3	kjar:3.0 (alias=kjar, group-id=com.example, artifact-id=sample-kjar, version=3.0)	129.0.3.1, 129.0.3.2, 129.0.3.3

If you send a request, the request is sent to **kie-server1 (129.0.1.2)**, **kie-server2 (129.0.2.3)**, and **kie-server3 (129.0.3.1)**.

If you send a second request, that request is sent to the next host of each KIE Server. For example, **kie-server1 (129.0.1.3)**, **kie-server2 (129.0.2.1)**, and **kie-server3 (129.0.3.2)**.

Smart Router has three components that you can modify to change this behavior:

ContainerResolver

The component responsible for finding the container id to use when interacting with servers.

RestrictionPolicy

The component responsible for disallowing Smart Router to use specific endpoints.

ConfigRepository

The component responsible for maintaining the Smart Router configuration. This is mainly related to the routing table.

IdentityService

The component responsible for allowing you to use your own identity provider. This is for KIE Server instances.

Smart Router uses the **ServiceLoader** utility to implement these components:

ContainerResolver

META-INF/services/org.kie.server.router.spi.ContainerResolver

RestrictionPolicy

META-INF/services/org.kie.server.router.spi.RestrictionPolicy

ConfigRepository

META-INF/services/org.kie.server.router.spi.ConfigRepository

IdentityService

META-INF/services/org.kie.server.router.identity.IdentityService

For example, for the above scenario, you can customize the **ContainerResolver** to make Smart Router search for the latest version of the KJAR process across all available KIE Servers and to always start with that process. This scenario would mean that each KIE Server hosts a single KJAR and each version will share the same alias.

Since Smart Router is an executable jar, to include extensions, you need to modify the command. For example:

```
java -cp LOCATION/router-ext-7.7.1.redhat-00002.jar:rhcam-7.8.0-smart-router.jar
org.kie.server.router.KieServerRouter
```

Once the service is started you will see log output stating the implementation that is used for the components:

```
Mar 01, 2017 1:47:10 PM org.kie.server.router.KieServerRouter <init>
INFO: KIE Server router repository implementation is InMemoryConfigRepository
Mar 01, 2017 1:47:10 PM org.kie.server.router.proxy.KieServerProxyClient <init>
INFO: Using 'LatestVersionContainerResolver' container resolver and restriction policy
'ByPassUserNotAllowedRestrictionPolicy'
Mar 01, 2017 1:47:10 PM org.xnio.Xnio <clinit>
INFO: XNIO version 3.3.6.Final
Mar 01, 2017 1:47:10 PM org.xnio.nio.NioXnio <clinit>
INFO: XNIO NIO Implementation Version 3.3.6.Final
Mar 01, 2017 1:47:11 PM org.kie.server.router.KieServerRouter start
INFO: KieServerRouter started on localhost:9000 at Wed Mar 01 13:47:11 CET 2017
```

CHAPTER 5. CONFIGURING QUARTZ TIMER SERVICE

When you run KIE Server in a cluster you can configure the Quartz timer service.

Before you configure a database on your application server, you must prepare the database for Quartz to create Quartz tables, which will hold the timer data, and the Quartz definition file.

Prerequisites

- A supported non-JTA data source is connected to your application server, for example a PostgreSQL data source.

Procedure

1. Create Quartz tables in your database to enable timer events to synchronize using the DDL script for your database.

The DDL script is available in the extracted supplementary ZIP archive in **QUARTZ_HOME/docs/dbTables**.



NOTE

Scripts containing the word **drop** such as **quartz_tables_drop_db2.sql** drop the Quartz table before creating it.

2. Create the Quartz configuration file **quartz-definition.properties** in the **JBOSS_HOME/MODE/configuration/** directory and add the following example content:

```
#=====
# Configure Main Scheduler Properties
#=====
org.quartz.scheduler.instanceName = jBPMClusteredScheduler
org.quartz.scheduler.instanceId = AUTO
#=====

# Configure ThreadPool
#=====

org.quartz.threadPool.class = org.quartz.simpl.SimpleThreadPool
org.quartz.threadPool.threadCount = 5
org.quartz.threadPool.threadPriority = 5
#=====

# Configure JobStore
#=====

org.quartz.jobStore.misfireThreshold = 60000
org.quartz.jobStore.class=org.quartz.impl.jdbcjobstore.JobStoreCMT
org.quartz.jobStore.driverDelegateClass=org.quartz.impl.jdbcjobstore.PostgreSQLDelegate
org.quartz.jobStore.useProperties=false
org.quartz.jobStore.dataSource=managedDS
org.quartz.jobStore.nonManagedTXDataSource=notManagedDS
org.quartz.jobStore.tablePrefix=QRTZ_
```

```

org.quartz.jobStore.isClustered=true
org.quartz.jobStore.clusterCheckinInterval = 20000
#=====

# Configure Datasources
#=====

org.quartz.dataSource.managedDS.jndiURL=jboss/datasources/psbpmsDS
org.quartz.dataSource.notManagedDS.jndiURL=jboss/datasources/quartzNotManagedDS
# Note the configured data sources that accommodate the two Quartz schemes at the very
end of the file.

```



IMPORTANT

The recommended interval for cluster discovery is 20 seconds and is set in the **org.quartz.jobStore.clusterCheckinInterval** attribute of the **quartz-definition.properties** file. Consider the performance impact on your system and modify the settings as necessary.

3. Provide the absolute path to your **quartz-definition.properties** file in the **org.quartz.properties** property.
4. Optional: To configure the number of retries and delay for the Quartz trigger, update the following system properties:
 - **org.jbpm.timer.quartz.retries** (default value is 5)
 - **org.jbpm.timer.quartz.delay** in milliseconds (default value is 1000)



NOTE

By default, Quartz requires two data sources:

- Managed data source to participate in the transaction of the process engine.
- Unmanaged data source to look up timers to trigger without any transaction handling

Red Hat Process Automation Manager business applications assume that the Quartz database (schema) will be co-located with Red Hat Process Automation Manager tables and therefore produce data sources used for transactional operations for Quartz.

The other (non transactional) data source must be configured but it should point to the same database as the main data source.

CHAPTER 6. ADDITIONAL RESOURCES

- *Installing and configuring Red Hat Process Automation Manager on Red Hat JBoss EAP 7.3*
- *Planning a Red Hat Process Automation Manager installation*
- *Deploying a Red Hat Process Automation Manager immutable server environment on Red Hat OpenShift Container Platform*
- *Deploying a Red Hat Process Automation Manager authoring environment on Red Hat OpenShift Container Platform*
- *Deploying a Red Hat Process Automation Manager freeform managed server environment on Red Hat OpenShift Container Platform*
- *Deploying a Red Hat Process Automation Manager environment on Red Hat OpenShift Container Platform using Operators*

APPENDIX A. VERSIONING INFORMATION

Documentation last updated on Monday, November 15, 2021.