Red Hat Middleware Management 7.0.TechPreview
Installing Red Hat Middleware Management with CloudForms

For Use with Red Hat Middleware Management

Red Hat Customer Content Services

Red Hat Middleware Management 7.0.TechPreview Installing Red Hat Middleware Management with CloudForms

---

For Use with Red Hat Middleware Management

Legal Notice

Copyright © 2017 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

This documentation describes the server and agent installation for Red Hat Middleware Management .

Table of Contents

CHAPTER 1. PREREQUISITES

Note

This release of Red Hat Middleware Management is a technical preview. Technology Previews provide early access to upcoming product innovations, letting you to test new features and provide feedback during the development process. Technology Preview releases are *not* intended for production use. For more information see the Red Hat Customer Portal.

This document provides instructions for installing Red Hat Middleware Management. The installation media is two Linux container images available from registry.access.redhat.com.

1.1. ARCHITECTURE OVERVIEW

Installing the Red Hat Middleware Management solution involves installing the following components:

- **Middleware management server**: A Red Hat JBoss Enterprise Application Platform (EAP) 7 based application that collects metrics and events from middleware servers and sends that data to the CloudForms console. The middleware management server also executes operations on middleware servers. This is delivered as a Linux container image.
- **Cassandra datastore**: Storage for middleware inventory and time series data. Only the middleware manager communicates with the metric store and it is not supported for any other use. This is delivered as a Linux container image.
- **Middleware management agent**: You install an agent on each JBoss middleware instance to be managed. Once installed and enabled, the agent can push metrics and events to the middleware management server.

Note

The middleware manager JAR files are built using the Java Developer Kit version 8 (JDK8). If you are using a version of Red Hat Enterprise Linux that does not include JDK8 (versions before RHEL 7), you will need to upgrade your JDK to version 8 in order to run the middleware manager server and agents.

1.2. INSTALLATION OVERVIEW

Before you begin installing Red Hat Middleware Management, you should have the following information:

- A non-root user to perform the middleware management server installation.
- A user with permission to write to the Cassandra data store (used for ongoing operations). This can be the same non-root user who performs the installation.

The installation workflow is as follows:

1. Install and configure Docker

2. Install and configure the Cassandra datastore
3. Install the Red Hat Middleware Management server
4. Add the middleware provider to CloudForms
5. Install the Red Hat Middleware Management agent on EAP servers
6. (Optional) Add nodes to Cassandra

1.3. INSTALLATION PREREQUISITES

The installation instructions for the Red Hat Middleware Management server assume that you already have the following in place:

- Red Hat Enterprise Linux 7.2 or higher or Red Hat Enterprise Atomic Host 7.2 or higher. For installation instructions, see the Red Hat Enterprise Linux installation guide.
- Red Hat CloudForms 4.2 or higher installed. For installation instructions, see the Red Hat CloudForms installation guide.
- Docker version 1.8 or higher.

1.3.1. Installing and Configuring Docker

1. Register your Linux machine and add the following repositories for Docker support.

```
subscription-manager register --username=<rhnuser> --password=<pwd>
subscription-manager list --available
subscription-manager attach --pool=<pool_id>
subscription-manager repos --enable=rhel-7-server-extras-rpms
```

2. Install Docker and the required dependencies.

```
yum install docker
```

3. Start the Docker service.

```
systemctl start docker
```

4. Enable the Docker service.

```
systemctl enable docker
```

CHAPTER 2. INSTALLING THE MIDDLEWARE MANAGEMENT SERVER

The Red Hat Middleware Management installation consists of two Linux container images.

Important

Do not perform the installation as the ROOT user; create a non-root user.

2.1. INSTALL AND CONFIGURE CASSANDRA

The middleware management services require a running Cassandra instance.

1. Run the Cassandra container and configure the Cassandra data location. By default the Cassandra container creates a volume under ***/opt/apache-cassandra/data***. This container directory automatically maps to a unique local directory which you can view using **`docker inspect <container-id>`**. You can use the **`-v`** option to map the Cassandra data volume to a specific host directory.

   Important

   `CASSANDRA_START_RPC` must be set to **`true`**. If this option is set to **`false`** the middleware manager services cannot connect to Cassandra.

   Note

   The **`Z`** option is necessary on Red Hat Enterprise Linux/Fedora based systems to correctly set the SELinux permissions.

   ```
   docker run --name mwmanager-cassandra -d -e
   CASSANDRA_START_RPC=true -v /var/mydatastore:/opt/apache-
   cassandra/data:Z registry.access.redhat.com/jboss-mm-7-tech-
   preview/middleware-manager-datastore:latest
   ```

2. Verify that the server is running.

   ```
   docker ps
   ```

   Result

   ```
   CONTAINER ID   IMAGE
   COMMAND                  CREATED          STATUS          PORTS
   NAMES
   34a75ba45be8   registry.access.redhat.com/jboss-mm-7-tech-
   ```

```
preview/middleware-manager-datastore:latest   "/docker-
entrypoint.s"   44 seconds ago   Up 44 seconds   7000-7001/tcp,
7199/tcp, 9042/tcp, 9160/tcp   mwmanager-cassandra
```

3. View the node information.

```
docker exec -it <container_id> /opt/apache-cassandra/bin/nodetool
info
```

2.1.1. Running Cassandra with a Non-default User

You can run the Cassandra container with an arbitrary user ID, however, that user must have permissions to write to the Cassandra data volume.

Note

The **Z** option is necessary on Red Hat Enterprise Linux/Fedora based systems to correctly set the SELinux permissions.

```
chown myuser:myuser /var/mydatastore
docker run --name mwmanager-cassandra -d -e CASSANDRA_START_RPC=true --
user $(id -u myuser) -v /var/mydatastore:/opt/apache-cassandra/data:Z
registry.access.redhat.com/jboss-mm-7-tech-preview/middleware-manager-
datastore:latest
```

2.2. STARTING THE MIDDLEWARE MANAGER SERVICES

1. Start the mwmanager-services and link it to the running Cassandra instance. By default the middleware manager container creates a volume under ***/var/opt/hawkular/data***. As with the Cassandra container, you can use the **-v** option to map the middleware manager data volume to a specific host directory.

Note

The **Z** option is necessary on Red Hat Enterprise Linux/Fedora based systems to correctly set the SELinux permissions.

```
docker run -d --link=mwmanager-cassandra \
-e CASSANDRA_NODES=mwmanager-cassandra \
-e HAWKULAR_BACKEND=remote \
-p 8080:8080 -p 8443:8443 -p 9990:9990 \
-v /var/opt/mwmanager-data:/var/opt/hawkular/data:Z
registry.access.redhat.com/jboss-mm-7-tech-preview/middleware-
manager:latest
```

2. Run the Middleware Manager container and tell it to connect to Cassandra. If desired, you can specify a combination of username/password for **HAWKULAR_USER** and **HAWKULAR_PASSWORD**. If you do not specify a user name and password, one will be automatically generated when you start the container.

```
docker run --name mwmanager  -d \
          -e HAWKULAR_BACKEND=remote   \
          -e CASSANDRA_NODES=mwmanager-cassandra \
          -e HAWKULAR_USE_SSL=true \
          -e HAWKULAR_USER=my_mwmanager_username \
          -e HAWKULAR_PASSWORD=my_mwmanager_password \
          -p 8080:8080 -p 8443:8443 -p 9990:9990 \
          --link=mwmanager-cassandra \
          registry.access.redhat.com/jboss-mm-7-tech-
preview/middleware-manager:latest
```

Table 2.1. Service Options

Name	Default	Description
CASSANDRA_NODES	myCassandra	The host name or IP of Cassandra
DB_TIMEOUT	-	If set, will wait for specified number of seconds for Cassandra to start and become ready before starting the middleware management services.
HAWKULAR_BACKEND	remote	Indicates that the middleware manager services connects to a Cassandra server that is installed remotely.
HAWKULAR_METRICS_TTL	17	

Name	Default	Description
HAWKULAR_USER		User name for the new user. If you do not specify a user name, one will be automatically generated. To view the generated user name, run the following command: **docker exec <containerID> bash -c 'echo "$HAWKULAR_USER"'**
HAWKULAR_PASSWORD		Password for the new user. If you do not specify a password, one will be automatically generated. To view the generated password, run the following command: **docker exec <containerID> bash -c 'echo "$HAWKULAR_PASSWORD"'**
HAWKULAR_USE_SSL	false	Whether to use secure socket layer (SSL) to establish an encrypted link between CloudForms and the middleware manager server.

3. Verify the middleware management installation by launching a browser and navigating to http://my_mwmanager_host:8080/. After a minute or two, you should see the middleware manager status page.


Middleware Manager for CloudForms
Red Hat CloudForms | EAP Agent Installer
Middleware Manager Services: Running 0.19.0.Final-redhat-2 (commit c10b8c1f0924125c2eaeb5ca81ac0b1a00dd20a2)
Middleware Manager Metrics: Running 0.21.3.Final-redhat-1 (commit bf1af3d5dfecacb6d947595513db038f1351c843)
Middleware Manager Alerts: Running 1.3.1.Final-redhat-2 (commit e02909a93e781b5f8f21575ddf5901ae30f2110a)
Middleware Manager Inventory: Running 1.1.0.Final (commit 0ff0967f0c9b896a1455b5799accd0a1a4d59f04)

CHAPTER 3. INITIAL CONFIGURATION

3.1. ENABLING SSL WITH MIDDLEWARE MANAGER

Secure Socket Layer (SSL) allows secure communications between browsers and web servers. The data sent is encrypted by one side, transmitted, and then decrypted by the other side before processing. Both the server and the browser encrypt all communication packets before sending out data.

You can provide a trusted third-party or a self-signed certificate to secure communications between the provider and CloudForms, or you can use an existing certificate managed by CloudForms to secure communications. There are three ways to enable SSL when you add a middleware provider to CloudForms. Each of these three methods assumes that you have set the environment variable **`HAWKULAR_USE_SSL`** to **`true`** when you started the middleware manager services.

1. Provide your public and private keys as two .PEM files located in ***/client-secrets/hawkular-services-private.key*** and ***/client-secrets/hawkular-services-public.pem***.
2. Provide both your public and private keys as a pksc12 file located in ***/client-secrets/hawkular-services.pkcs12***.
3. Provide an auto-generated self-signed certificate in .PEM format when you add the Middleware Provider in CloudForms.

> **Warning**
>
> Using a self-signed SSL certificate to create a keystore is not intended for production environments. For production environments or where SSL encrypted communication is required, you must use a SSL certificate that is purchased from a verified Certificate Authority.

For instructions on how to generate a self-signed certificate with Cloudforms, see the CloudForms Hardening Guide.

For instructions on how to make your self-signed certificate trusted with Cloudforms, see the CloudForms Hardening Guide.

3.2. CONFIGURING MIDDLEWARE MANAGER WITH OPENSHIFT

If you are deploying Middleware Manager in the Red Hat OpenShift Container Platform to monitor Red Hat JBoss Enterprise Application Platform (EAP), note that if you update the Middleware Manager credentials (via the Middleware Manager pod environment variables), you must also update the values for the following credentials for each EAP container so that the EAP containers can reconnect to the Middleware Manager using the new credentials:

- **`AB_HAWKULAR_REST_USER=`**
- **`AB_HAWKULAR_REST_PASSWORD=`**

Note that when you change pod environment variables, OpenShift will restart the pod and all running containers within that pod.

Caution

If the EAP **`AB_HAWKULAR_REST_*`** credentials do not match the Middleware Manager credentials, the EAP servers will be inaccessible to the Middleware Manager.

3.3. ADDING A MIDDLEWARE PROVIDER TO CLOUDFORMS

The middleware provider extends CloudForms management capabilities to JBoss Middleware application containers running in managed virtual machines, hosts, and Linux containers. The provider delivers inventory, events, metrics, and power operations. Middleware management in CloudForms is a provider based on the Hawkular open source project. When feature complete, the middleware provider will replace the current Red Hat middleware management offering, JBoss Operations Network.

1. Log in to the CloudForms Management Engine as a user who has permissions to add providers. The default user is **`admin`**, password **`smartvm`**.
2. Navigate to **Middleware** → **Providers**.
3. Click  (**Configuration**), then click  (**Add a New Middleware Provider**).
4. Enter a **Name** for the provider, for example, Middleware Manager.
5. From the **Type** list, select **Hawkular**.
6. Accept the default **Zone**.
7. Under **Endpoints**, configure the following for the middleware provider:

    a. Select a **Security protocol** method to specify how to authenticate to the provider. In order to use SSL, the middleware manager server must have been started with the **`HAWKULAR_USE_SSL`** option set to **`true`**.

    Note

    The only supported security protocols for the current release are **SSL without validation** and **Non-SSL**.

    - **SSL** (SSL with validation) – Authenticate to the provider securely using a trusted certificate authority. This requires that you have already configured your public and private keys in the ***/client-secrets*** directory as either two .PEM files or as a single .pkcs12 file.
    - **SSL trusting custom CA** - Authenticate to the provider with a self-signed certificate. For this option, copy your certificate's text to the **Trusted CA Certificates** field in .PEM format.
    - **SSL without validation** – Authenticate to the provider insecurely (not recommended).
    - **Non-SSL** - Select if you do not want to use SSL.

b. Enter the **Hostname** or IPv4 or IPv6 address of the machine where you installed the middleware manager.

c. Enter the **API Port** of the middleware manager. The default is 8080.

d. Enter the **User Name** used to start the middleware manager. This should match the **`HAWKULAR_USER`**.

e. Enter the **Password** used to start the middleware manager. This should match the **`HAWKULAR_PASSWORD`**.

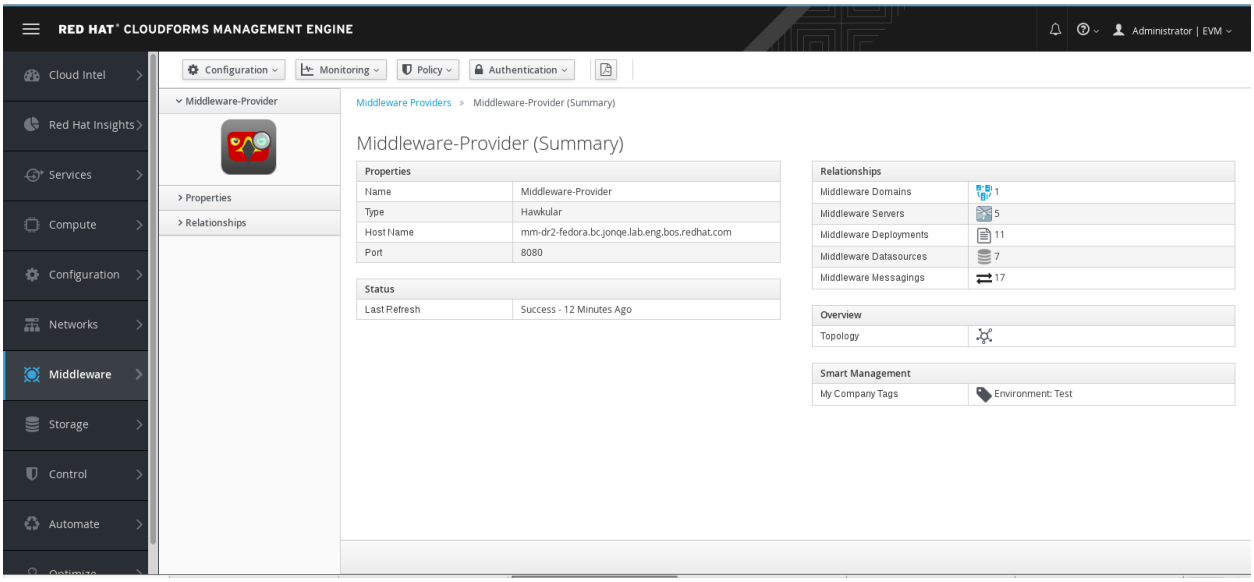
f. Reenter the password in the **Confirm Password** field.

g. Click **Validate** to confirm that the user has the proper credentials.

8. Click **Add**.

9. Click  (**Configuration**), then click  (**Refresh Items and Relationships**).

Red Hat CloudForms displays the Summary screen.



3.4. INSTALLING THE MIDDLEWARE MANAGEMENT AGENT

The middleware manager requires that you install an agent to enable it to perform monitoring and management functions. Once installed on a server or container, the agent can push metrics and events to the middleware manager.

3.4.1. Agent Overview

In order to instrument an Enterprise Application Platform (EAP) server, you must install an agent on the EAP server. The agent lets the middleware manager execute power operations on the EAP server and perform other operations such as database driver creation and application deployment.

To meet the needs of different server platforms, there are two middleware manager agents:

- **Middleware Manager EAP Agent** - You can use the EAP agent to monitor EAP application servers as well as applications running within those application servers. The agent runs embedded within an EAP server as a subsystem extension, and can be run in standalone or domain mode.

Note

EAP 6 does not support custom subsystem extensions, therefor you cannot run the agent in domain mode on EAP 6.

The EAP agent can collect inventory and metrics from the native EAP management interface (DMR API) as well as JMX (via Jolokia if monitoring remote application servers). The EAP agent is available via the EAP 6 and EAP 7 RPM streams, and as a zip file that you can download from the middleware manager status page. You use the EAP agent if you are deploying an EAP server on bare metal or in a virtual machine.

- **Middleware Manager Java Agent** - You can use the Java agent to monitor any Java Virtual Machine (JVM) or Java based application, including non-EAP Java applications. The Java agent can read and collect JMX metric data. The Java agent is embedded in the middleware manager container image, and is part of the EAP 6 and EAP 7 container images. You use the Java agent if you are deploying EAP in a container, or deploying a non-EAP Java application in a container.

Note

EAP 7 does not support running the Java agent in domain mode.

3.4.2. Installing the EAP Agent via Zip

The EAP agent zip installer is available from the middleware manager status page.

1. Launch a browser and go to http://my_mwmanager_host:8080/.
2. Click the **EAP Agent Installer** link.
3. In the Authentication Required pop-up, enter the credentials for the middleware manager user (the **`HAWKULAR_USER`** and **`HAWKULAR_PASSWORD`**).

   The middleware management server downloads a file named **`hawkular-wildfly-agent-installer.jar`** to your local machine.

3.4.2.1. Install the Agent in Enterprise Application Platform

Note

The agent installer requires the EAP server to be down during the agent installation process.

After you have downloaded the **`.jar`** file, use one of the following commands to copy the file to the EAP server (if it is on a different machine) and install the agent. Select the command that matches the mode of your EAP server.

Standalone mode

```
java -jar ~Downloads/hawkular-wildfly-agent-installer.jar
```

```
   --target-location=/opt/jboss-eap-7.0/
   --server-url=http://my_mwmanager_host:8080
   --username=my_mwmanager_username
   --password=my_mwmanager_password
```

Domain mode

```
java -jar ~Downloads/hawkular-wildfly-agent-installer.jar
   --target-location=/opt/jboss-eap-7.0/
   --target-config=/opt/jboss-eap-7.0/domain/configuration/host.xml
   --server-url=http://my_mwmanager_host:8080
   --username=my_mwmanager_username
   --password=my_mwmanager_password
```

3.4.2.2. Verify the Agent Installation in CloudForms

1. Start or restart the newly instrumented EAP server.
2. In CloudForms, select **Middleware**, then **Providers** and select the middleware management provider that you previously configured.
3. On the Summary screen, in the **Relationships** table, click the **Middleware Servers** icon to view the instrumented servers.

Note

The new server may take a few moments to appear.

3.4.3. Installing the EAP Agent via RPM from EAP 6

The middleware management agent is available via the JBoss EAP 6 RPM stream.

3.4.3.1. Installation Prerequisites for EAP 6 Agent

These installation instructions for the EAP agent assume that you have already completed the following tasks:

1. Configured the EAP 6 repository.
   - For configuration instructions via either Red Hat Subscription Manager or Red Hat Network Classic, see the Red Hat JBoss Enterprise Application Platform 6 installation instructions.
   - To configure the repository manually, create a new configuration file called **`/etc/yum.repos.d/jboss-eap.repo`**.

   /etc/yum.repos.d/jboss-eap.repo

   ```
   [jboss-eap6]
   name=JBoss EAP 6.4
   ```

```
baseurl=http://download.devel.redhat.com/released/jboss/eap6/6
.4.13/composes/JBEAP-6.4.13-RHEL-7/Server/x86_64/os/
gpgcheck=0
```

2. Installed EAP 6 using the following command:

```
yum groupinstall jboss-eap6
```

For complete installation and configuration instructions, see the Red Hat JBoss Enterprise Application Platform 6 installation instructions.

3. Stopped the EAP server if it is running.

3.4.3.2. Installing the Agent on EAP 6

Note

The agent installer requires the EAP server to be down during the agent installation process. Also, the middleware manager JAR files are built using the Java Developer Kit version 8 (JDK8). If you are using a version of Red Hat Enterprise Linux that does not include JDK8 (versions before RHEL 7), you will need to upgrade your JDK to version 8 in order to run the middleware manager server and agents.

1. Install the required EAP module.

```
yum install middleware-manager-agent-eap6
```

2. Configure the server:

```
$ export JBOSS_HOME=/usr/share/jbossas/
$ java -jar ${JBOSS_HOME}/bin/hawkular-wildfly-agent-
installer.jar \
       --target-location=${JBOSS_HOME} \
       --server-url=http://${HAWKULAR_SERVER}:${HAWKULAR_PORT} \
       --username=jdoe \
       --password=password \
       --config-only=true \
```

3.4.3.3. Verify the Agent Installation in CloudForms

1. Start or restart the newly instrumented EAP 6 server.

```
systemctl start jbossas
```

2. In CloudForms, select **Middleware**, then **Providers** and select the middleware management provider that you previously configured.

3. On the Summary screen, in the **Relationships** table, click the **Middleware Servers** icon to view the instrumented servers. The new server may take a few moments to appear.

Note

If you change the Middleware Manager credentials, you must also change the AB_HAWKULAR_REST_* credentials for EAP, otherwise the Middleware Manager will not be able to access the EAP server.

3.4.4. Installing the EAP Agent via RPM from EAP 7

The middleware management agent is available via the JBoss EAP 7 RPM stream.

3.4.4.1. Installation Prerequisites for EAP 7 Agent

These installation instructions for the EAP agent assume that you have already completed the following tasks:

1. Configured the EAP 7 repository.
   - For configuration instructions via Red Hat Subscription Manager, see the Red Hat JBoss Enterprise Application Platform 7 installation instructions.
   - To configure the repository manually, create a new configuration file called **`/etc/yum.repos.d/jboss-eap.repo`**.

   /etc/yum.repos.d/jboss-eap.repo

   ```
   [jboss-eap7]
   name=JBoss EAP 7
   enabled=1
   gpgcheck=0
   baseurl=http://download.devel.redhat.com/released/jboss/eap7/7
   .0.4/composes/JBEAP-7.0.4-RHEL-7/Server/x86_64/os/
   ```

2. Installed EAP 7 using the following command:

   ```
   yum groupinstall "JBoss EAP 7"
   ```

   For complete installation and configuration instructions, see the Red Hat JBoss Enterprise Application Platform 7 installation instructions.

3. Stopped the EAP server if it is running.

3.4.4.2. Installing the Agent on EAP 7

Note

The agent installer requires the EAP server to be down during the agent installation process. Also, the middleware manager JAR files are built using the Java Developer Kit version 8 (JDK8). If you are using a version of Red Hat Enterprise Linux that does not include JDK8 (versions before RHEL 7), you will need to upgrade your JDK to version 8 in order to run the middleware manager server and agents.

1. Install the required EAP module.

```
yum install eap7-middleware-manager-agent
```

2. Configure the server:

```
export JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly/
export HAWKULAR_SERVER=<hawkular-server>
export HAWKULAR_PORT=8080
java -jar ${JBOSS_HOME}/bin/hawkular-wildfly-agent-installer.jar
\
       --target-location=${JBOSS_HOME} \
       --server-url=http://${HAWKULAR_SERVER}:${HAWKULAR_PORT} \
       --username=jdoe \
       --password=password \
       --config-only=true
```

3.4.4.3. Verify the Agent Installation in CloudForms

1. Start or restart the newly instrumented EAP 7 server.

```
systemctl start eap7-standalone
```

2. In CloudForms, select **Middleware**, then **Providers** and select the middleware management provider that you previously configured.
3. On the Summary screen, in the **Relationships** table, click the **Middleware Servers** icon to view the instrumented servers.The new server may take a few moments to appear.

Note

If you change the Middleware Manager credentials, you must also change the AB_HAWKULAR_REST_* credentials for EAP, otherwise the Middleware Manager will not be able to access the EAP server.

3.4.5. Un-installing the Middleware Management Agent

Warning

Uninstalling a JBoss EAP agent installation that was installed using the RPM method is not recommended.

Because of the nature of RPM package management, it cannot be guaranteed that all installed packages and dependencies will be completely removed, or that the system will not be left in an inconsistent state caused by missing package dependencies.

3.5. MANAGING EAP CONTAINERS

You can obtain container images for EAP 6 and EAP 7 that already contain the middleware manager Java agent from the Red Hat Container Catalog. Note that these EAP containers are immutable, meaning that neither the middleware manager nor the agent can perform power or deployment operations on the containers.

3.5.1. Starting the EAP Container

You use the **`docker run`** command to start your EAP container.

Note

The EAP container should be started in standalone mode. Domain mode is not supported in the technical preview.

```
docker run -d -e AB_HAWKULAR_REST_URL="http://$<middleware-services-
URL>/" -e AB_HAWKULAR_REST_USER="jdoe" -e
AB_HAWKULAR_REST_PASSWORD="password" registry.access.redhat.com/jboss-
eap-7-tech-preview/eap70
```

For more information about EAP container images, see the Using the Red Hat JBoss Enterprise Application Platform Docker Image guide.

3.5.2. Configuring the Java Agent

You configure the Java agent by creating a YAML file. Because you pass the name and location of the configuration file as a command line argument, you can give the file any name. But the recommended location is the EAP **`standalone/configuration`** directory.

The configuration model for the Java agent closely mimics that of the EAP **`standalone.xml`** configuration settings. For more information about the **`standalone.xml`** file, see the Red Hat JBoss Enterprise Application Platform Getting Started Guide.

For an example configuration that instructs the Java agent to monitor a EAP 7.0 or 7.1 server, see this sample configuration file.

For an example configuration that instructs the Java agent to monitor a simple JMX application, see this sample configuration file.

3.5.3. Starting the Java Agent

To start the Java agent, you pass in **`-javaagent`** JVM command line arguments to your Java or JVM application start command, including the location of the JAR file and the YAML file. The delay option tells the agent to delay its start up the given number of seconds. This gives your main application time to start up before the agent begins monitoring it.

```
-javaagent:<path-to-jar>=config=<path-to-config-file>,delay=60
```

For example:

```
-javaagent:hawkular-javaagent.jar=config=javaagent-config.yaml,delay=60
```

3.6. CASSANDRA CLUSTER SETUP

3.6.1. Adding Cassandra Nodes

If you need to expand the Cassandra cluster on the same machine, you can launch a new Cassandra container and configure it to point to the seed container. Ensure that you provide a valid seed container so the new node can bootstrap properly.

1. Launch the new Cassandra container.

Note

If the cluster needs to run on a separate machine, you must specify the option **`-e CASSANDRA_BROADCAST_ADDRESS=<public-ip>`** on each node that is launched, where **`public-ip`** is the IP of the machine on the network. It is not possible to combine the approach of having multiple nodes on the same box and nodes on a remote box for one cluster.

```
docker run -d -e CASSANDRA_SEEDS=mwmanager-cassandra -d \
              --link mwmanager-cassandra  \
              registry.access.redhat.com/jboss-mm-7-tech-
preview/middleware-manager-datastore:latest
```

2. Verify the container has launched using the **`docker ps`** command.

3. Verify the status of the node by running the **`nodetool`** command inside the container.

```
docker exec -it mwmanager-cassandra /opt/apache-
cassandra/bin/nodetool status

Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load       Tokens       Owns (effective)  Host
ID                              Rack
UN  172.17.0.2  490.35 KB  256          50.2%
ead3a0ee-b040-4873-8b62-aa700d02b0c1  rack1
UN  172.17.0.4  312.28 KB  256          49.8%
8e4a73eb-5545-48e1-9464-ef2728f8852e  rack1
```

4. After all nodes appear UP you should run the cleanup process in each of the other nodes of the cluster. This operation removes all unnecessary keys that don't belong to the node itself. You can perform this operation using **`nodetool cleanup`** inside each container.

Note

The cleanup process needs temporary disk space (proportional to the amount of data stored) and is an I/O intensive operation, so it should be postponed to lower usage hours.

```
docker exec -it <container_id> /opt/apache-cassandra/bin/nodetool
cleanup
```

Each time you want to add a new node to the cluster, you should repeat this entire process.

3.6.2. Removing a Cassandra Node.

1. Select the node to be removed from the cluster. You can obtain a list of nodes by running **`docker ps`**.
2. After you have selected the node, check whether the node is up or down using **`nodetool status`**.
3. Run the decommission process inside the container. This process will move data to the other nodes and replicate the appropriate data.

```
  docker exec -it <my_container_id> /usr/bin/nodetool decommission
```

4. You can monitor the progress of the process using **`nodetool netstats`**.

```
docker exec -it <container_id> /opt/apache-cassandra/bin/nodetool
netstats
```

```
Mode: DECOMMISSIONED
Not sending any streams.
Read Repair Statistics:
```

5. Once the process finishes, you can delete the container.

```
docker stop <my_container_id>
docker rm <my_container_id>
```

or

```
docker rm --force my_container_id
```

3.6.3. Replacing a Cassandra Node

If something goes wrong with one node, you can replace it with a new one.

1. Get the status of the cluster using **`nodetool status`**.

```
docker exec -it mwmanager-cassandra /opt/apache-
cassandra/bin/nodetool status

=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load       Tokens       Owns (effective)  Host
ID                              Rack
UN  172.17.0.3  179.25 KiB  256          67.6%
```

```
3dc29aa0-5cb7-4352-a45d-72600f87ee48  rack1
DN  172.17.0.2  166.47 KiB  256          63.4%
b73c89e5-e2e1-470e-874b-f926b7243b49  rack1
UN  172.17.0.4  83.71 KiB  256          69.0%
fe1e6949-1fc7-496a-a572-a3416b47d16f  rack1
```

2. If there is a dead node, get the IP address of that node.
3. Start a new node container using the following command, where **`node_replaced_ip`** is the IP of the node you want to replace.

```
docker run -d -e JVM_OPTS="-Dcassandra.replace_address=
<node_replaced_ip>" \
           -e CASSANDRA_SEEDS=mwmanager-cassandra\
           -e CASSANDRA_START_RPC=true \
           --link mwmanager-cassandra \
           registry.access.redhat.com/jboss-mm-7-tech-
preview/middleware-manager-datastore:latest
```

4. Run **`nodetool status`** again to see if the node was replaced.
5. After the new node finishes the bootstrap process, you can remove the old node.

Revised on 2017-07-05 15:45:25 EDT