Deploying JBoss EAP on Amazon Web Services

Instructions for using Red Hat JBoss Enterprise Application Platform on Amazon Web Services, including configuring high availability.
Instructions for using Red Hat JBoss Enterprise Application Platform on Amazon Web Services, including configuring high availability.
Abstract

This document provides information about deploying Red Hat JBoss Enterprise Application Platform on Amazon EC2.
Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION .................................................. 4

MAKING OPEN SOURCE MORE INCLUSIVE ................................................................. 5

CHAPTER 1. ABOUT AMAZON EC2 ................................................................. 6
  About Red Hat Cloud Access 6
  Red Hat Cloud Access Features 6
  Supported Amazon EC2 Instance Types 6
  Supported Red Hat AMIs 7

CHAPTER 2. LAUNCHING A JBOSS EAP INSTANCE ON AMAZON EC2 CONSOLE .............. 8

CHAPTER 3. LAUNCHING A NON-CLUSTERED JBOSS EAP INSTANCE ....................... 9
  Prerequisites 9
  Launch a Non-Clustered JBoss EAP Instance 9

CHAPTER 4. LAUNCHING NON-CLUSTERED MANAGED DOMAIN .................................. 11
  4.1. LAUNCH AN INSTANCE TO SERVE AS A DOMAIN CONTROLLER .................... 11
      Prerequisite 11
      Launch a Non-Clustered JBoss EAP Instance 11
  4.2. LAUNCH ONE OR MORE INSTANCES TO SERVE AS HOST CONTROLLERS ............ 12
      For Domain Controller Instance 12

CHAPTER 5. LAUNCHING CLUSTERED JBOSS EAP .................................................. 14
  5.1. LAUNCH CLUSTERED JBOSS EAP AMIS (WITHOUT MOD_CLUSTER AND VPC) ............ 14
      For Domain Controller Instance 15
      For Host Controller Instance 15
  5.2. LAUNCH CLUSTERED JBOSS EAP AMIS (WITH MOD_CLUSTER AND VPC) .............. 16
      5.2.1. To launch clustered AMIs with VPC and mod_cluster 17

CHAPTER 6. TROUBLESHOOTING .............................................................................. 19
  6.1. ABOUT TROUBLESHOOTING AMAZON EC2 ..................................................... 19
  6.2. DIAGNOSTIC INFORMATION .......................................................................... 19

APPENDIX A. REFERENCE MATERIAL ....................................................................... 20
  A.1. AMAZON EC2 AMIS FOR RED HAT CLOUD ACCESS PROGRAM ....................... 20
      AMIs for Platform/JDK Combinations: 20
      Maintenance of AMIs 20
      Scenario 1 (Supported) 20
  A.2. EXAMPLE CONFIGURATION FILES AND DEPLOYMENTS ................................. 20
  A.3. SYSTEM PATHS ................................................................................................. 20
      Service Configuration Files: 21
      JBoss EAP Home: 21
      JBoss EAP Configuration Locations: 21
      JBoss EAP Example Configuration Locations: 21
  A.4. LAUNCHING JBOSS EAP ON AMAZON EC2 USING A SCRIPT ......................... 21
  A.5. CONFIGURING JBOSS EAP SUBSYSTEMS TO WORK ON CLOUD PLATFORMS ......... 22
      A.5.1. Web Services 22
      A.5.2. Messaging 22
      A.5.3. Remoting Configuration for High Availability 22
  A.6. EXAMPLE USER DATA FOR CLUSTERED JBOSS EAP Instances ....................... 23
      Example: File for Standalone Mode on RHEL6/7 23
      Example: File for Starting a Clustered Domain Instance (Domain Controller) 24
Example: File for Starting a Clustered Domain Instance (Host Controller)
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments. Follow the steps in the procedure to learn about submitting feedback on Red Hat documentation.

Prerequisites

- Log in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in Multi-page HTML format.

Procedure

1. Click Feedback to see existing reader comments.

   **NOTE**
   
   The feedback feature is enabled only in the Multi-page HTML format.

2. Highlight the section of the document where you want to provide feedback.

3. In the prompt menu that displays near the text you selected, click Add Feedback.
   A text box opens in the feedback section on the right side of the page.

4. Enter your feedback in the text box and click Submit.
   You have created a documentation issue.

5. To view the issue, click the issue tracker link in the feedback view.
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
CHAPTER 1. ABOUT AMAZON EC2

Amazon Elastic Compute Cloud (Amazon EC2), a service operated by amazon.com, provides customers with a customizable virtual computing environment. With this service, an Amazon Machine Image (AMI) can be booted to create a virtual machine or instance. Users can install the software they require on an instance and are charged according to the capacity used. Amazon EC2 is designed to be flexible and allows users to quickly scale their deployed applications.

See the Amazon Web Services website for more information.

About Amazon Machine Images

An Amazon Machine Image (AMI) is a template for an EC2 virtual machine instance. Users create EC2 instances by selecting an appropriate AMI to create the instance from. The primary component of an AMI is a read-only filesystem that contains an installed operating system as well as other software. Each AMI has different software installed for different use cases. Amazon EC2 includes many AMIs that both Amazon Web Services and third parties provide. Users can also create their own custom AMIs.

About Red Hat Cloud Access

Red Hat Cloud Access is a Red Hat subscription feature that provides support for JBoss EAP on Red Hat certified cloud infrastructure providers, such as Amazon EC2 and Microsoft Azure. Red Hat Cloud Access allows you to move your subscriptions between traditional servers and public cloud-based resources in a simple and cost-effective manner.

You can find more information about Red Hat Cloud Access on the Customer Portal.

Red Hat Cloud Access Features

Membership in the Red Hat Cloud Access program provides access to supported private Amazon Machine Images (AMIs) created by Red Hat.

The Red Hat AMIs have the following software pre-installed and fully supported by Red Hat:

- Red Hat Enterprise Linux
- JBoss EAP
- Product updates with RPMs using Red Hat Update Infrastructure

Each of the Red Hat AMIs is only a starting point, requiring further configuration to the requirements of your application.

IMPORTANT

Red Hat Cloud Access does not currently provide support for the full-ha profile, in either standalone instances or a managed domain.

NOTE

For information about installing Red Hat JBoss Operations Network, see the Red Hat JBoss Operations Network Installation Guide.

For information about configuring Red Hat JBoss Operations Network, see Configuring JBoss ON Servers, Agents, and Storage Nodes.

Supported Amazon EC2 Instance Types
Red Hat Cloud Access supports the following Amazon EC2 instance types. See Amazon Elastic Compute Cloud User Guide for Linux Instances for more information about each instance.

### Table 1.1. Supported Amazon EC2 Instance Types

<table>
<thead>
<tr>
<th>Instance Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Instance</td>
<td>Standard Instances are general purpose environments with a balanced memory-</td>
</tr>
<tr>
<td></td>
<td>to-CPU ratio. The smallest instance types available, which are capable of handling</td>
</tr>
<tr>
<td></td>
<td>JBoss EAP, are <strong>t2.small</strong> and <strong>m3.medium</strong>.</td>
</tr>
<tr>
<td>High Memory Instance</td>
<td>High Memory Instances have more memory allocated to them than Standard</td>
</tr>
<tr>
<td></td>
<td>Instances. High Memory Instances are suited for high throughput applications</td>
</tr>
<tr>
<td></td>
<td>such as databases or memory caching applications. Smallest available and</td>
</tr>
<tr>
<td></td>
<td>supported instance type is <strong>r3.large</strong>.</td>
</tr>
<tr>
<td>High CPU Instance</td>
<td>High CPU Instances have more CPU resources allocated than memory and are</td>
</tr>
<tr>
<td></td>
<td>suited for relatively low throughput but CPU intensive applications.</td>
</tr>
<tr>
<td></td>
<td>Smallest available and supported instance types are <strong>c3.large</strong> and <strong>c4.large</strong>.</td>
</tr>
</tbody>
</table>

**IMPORTANT**

The instance types Micro (**t2.micro**) and Nano (**t2.nano**) are not suitable for deployment of JBoss EAP. JBoss EAP 7.4 AMIs are built from a snapshot, which requires a volume of at least 10 GB. This can be set in EC2 console when creating the instance. If the volume assigned is too small, the instance creation will fail.

**Supported Red Hat AMIs**

The supported Red Hat AMIs can be identified by their names. JBoss EAP AMIs are named using the following syntax:

```
RHEL-osversion-_HVM_GA-JBEAP-version-creationdate-arch-1-Access2-GP2
```

- **version** is the version number of JBoss EAP installed in the AMI. Example **6.3**.
- **osversion** is the version number of Red Hat Enterprise Linux installed in the AMI. Example **6.2**.
- **arch** is the architecture of the AMI. This will be **x86_64** or **i386**.
- **creationdate** is the date that the AMI was created in the format of YYYYMMDD. Example **20160315**.

Example AMI name: `RHEL-7.3_HVM_GA-JBEAP-7.1.0_-20170703-x86_64-1-Access2-GP2`.
CHAPTER 2. LAUNCHING A JBOSS EAP INSTANCE ON AMAZON EC2 CONSOLE

You can launch a JBoss EAP instance on Amazon EC2 using the EC2 console.

You can also launch an instance using the AWS Command Line Interface. See AWS CLI for more information about AWS CLI.

1. Open the Amazon EC2 console.

2. From the Amazon EC2 console, click AMIs.

3. Search for jbeap AMI in Private images and select the AMI. For example, RHEL-7.3_HVM_GA-JBEAP-7.1.0_-20170703-x86_64-1-Access2-GP2.

4. Choose an instance type. See Supported Amazon EC2 Instance Types for more information on supported Amazon EC2 instance types.

5. In the Configure Instance Details section, configure the instance settings.

6. In the Advanced Details section, User data box, you can paste the sample script to run JBoss EAP when the instance is launched. See Launching JBoss EAP on Amazon EC2 Using Script for information about the sample script.

   NOTE
   If required, you can specify the storage, tag the instance, and configure the security group details.

7. Click Review and Launch. This takes you directly to the Review Instance Launch page.

8. Click Launch to choose a key pair and launch the instance.

   NOTE
   If you have not selected a key pair, you need to specify a key pair before you launch an instance.
CHAPTER 3. LAUNCHING A NON-CLUSTERED JBOSS EAP INSTANCE

This chapter lists the steps to launch a non-clustered instance of JBoss EAP on a Red Hat Amazon Machine Image (AMI).

Prerequisites

- A suitable Red Hat AMI. See Supported Red Hat AMIs for more information.
- A pre-configured Security Group that allows incoming requests on at least ports 22, 8080, and 9990.

Launch a Non-Clustered JBoss EAP Instance

NOTE

You can connect to an EC2 instance through ssh as the ec2-user user. If you need administrative privileges, you can change to root user later. For example,

```
$ ssh -l ec2-user ${INSTANCE_PUBLIC_IP}
...
$ sudo su -
```

- Launch the Red Hat AMI instance.
  A non-clustered instance of JBoss EAP has been configured and launched on a Red Hat AMI.
- To configure JBoss EAP, you can pass arguments to the service directly. Some arguments may not be handled in this way. The location of the service configuration files is:
  - RHEL 6: /etc/sysconfig/eap7-standalone
  - RHEL 7: /etc/opt/rh/eap7/wildfly/eap7-standalone.conf

See System Paths for more details about system paths.

NOTE

- For complex configuration, you can either use the standalone.conf file in the JBoss EAP bin directory: /opt/rh/eap7/root/usr/share/wildfly/bin/, or you can start the JBoss EAP service and configure the server using CLI. The script can be found in the bin directory. Then, reload the configuration.
- You must regularly run the yum -y update to apply security fixes and enhancements.

- To start JBoss EAP on RHEL 6, run the following command:

  ```$ service eap7-standalone start```

- To start JBoss EAP on RHEL 7, run the following command:

  ```$ systemctl start eap7-standalone```
To stop JBoss EAP on RHEL 6, run the following command:

```
$ service eap7-standalone stop
```

To stop JBoss EAP on RHEL 7, run the following command:

```
$ systemctl stop eap7-standalone
```

**NOTE**

The `systemctl` command is relevant to only RHEL 7.

**NOTE**

If you want to bind JBoss EAP to a different IP address, add the following line in the `/etc/opt/rh/eap7/wildfly/eap7-standalone.conf` file on RHEL 7. The internal IP address is translated into a public IP address by EC2.

```
WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS
```
CHAPTER 4. LAUNCHING NON-CLUSTERED MANAGED DOMAIN

4.1. LAUNCH AN INSTANCE TO SERVE AS A DOMAIN CONTROLLER

This topic lists the steps to launch a non-clustered JBoss EAP managed domain on a Red Hat Amazon Machine Image (AMI).

Prerequisite

- A suitable Red Hat AMI. See Supported Red Hat AMIs for more information.

Launch a Non-Clustered JBoss EAP Instance

**NOTE**

You can connect to an EC2 instance through `ssh` as the `ec2-user` user. If you need administrative privileges, you can change to `root` user later. For example,

```
$ ssh -l ec2-user ${INSTANCE_PUBLIC_IP}
...
$ sudo su -
```

- Launch the Red Hat AMI instance.
  A non-clustered instance of JBoss EAP has been configured and launched on a Red Hat AMI.

- To configure JBoss EAP, you can pass arguments to the service directly. Some arguments may not be handled in this way. The location of the service configuration files is:

  - RHEL 6: `/etc/sysconfig/eap7-domain`
  - RHEL 7: `/etc/opt/rh/eap7/wildfly/eap7-domain.conf`

See System Paths for more details about system paths.

See Configuring JBoss EAP Subsystems to Work on Cloud Platforms for information about configuring JBoss EAP subsystems for Amazon EC2.

**NOTE**

- For complex configuration, you can either use the `domain.conf` file in the JBoss EAP `bin` directory: `/opt/rh/eap7/root/usr/share/wildfly/bin/`, or you can start the JBoss EAP service and configure the server using the management CLI. The script can be found in the `bin` directory. Then, reload the configuration.

- You must regularly run the `yum -y update` to apply security fixes and enhancements.

- To start JBoss EAP on RHEL 6, run the following command:

  `$ service eap7-domain start`

To start JBoss EAP on RHEL 7, run the following command:
$ systemctl start eap7-domain

- To stop JBoss EAP on RHEL 6, run the following command:

  $ service eap7-domain stop

To stop JBoss EAP on RHEL 7, run the following command:

  $ systemctl stop eap7-domain

NOTE

The `systemctl` command is relevant to only RHEL 7.

NOTE

If you want to bind JBoss EAP to a different IP address, add the following line in the `/etc/opt/rh/eap7/wildfly/eap7-domain.conf` file on RHEL 7. The internal IP address is translated into a public IP address by EC2.

  WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS

4.2. LAUNCH ONE OR MORE INSTANCES TO SERVE AS HOST CONTROLLERS

This topic lists the steps to launch one or more instances of JBoss EAP to serve as non-clustered host controllers on a Red Hat AMI.

Configure and launch the non-clustered domain controller. Refer to Launch an Instance to Serve as a Domain Controller.

See System Paths for more details about system paths.

See Configuring JBoss EAP Subsystems to Work on Cloud Platforms for information about configuring JBoss EAP subsystems for Amazon EC2.

For Domain Controller Instance

For a managed domain running on Amazon EC2, in addition to static domain controller discovery, host controllers can dynamically discover a domain controller using the Amazon Simple Storage (Amazon S3) system. In particular, host controllers and the domain controller can be configured with information needed to access an Amazon S3 bucket.

Using this configuration, when a domain controller is started, it writes its contact information to an S3 file in the bucket. Whenever a host controller attempts to contact the domain controller, it gets the domain controller’s contact information from the S3 file.

For example, it is common for an Amazon EC2 instance’s IP address to change when it is stopped and started. In this scenario, if the domain controller’s contact information changes, the host controllers need not be reconfigured. The host controllers are able to get the domain controller’s new contact information from the S3 file.

See Example User Data for Clustered JBoss EAP Instances for more information about sample script for user data.
The manual domain controller discovery configuration is specified using the following properties:

- **access-key**: The Amazon AWS user account access key.
- **secret-access-key**: The Amazon AWS user account secret access key.
- **location**: The Amazon S3 bucket to be used.

1. Copy the `domain-ec2.xml` file from 
   `/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs` to the JBoss EAP configuration directory.

2. Set the following variables in the appropriate service configuration file:

   ```
   WILDFLY_SERVER_CONFIG=domain-ec2.xml
   WILDFLY_HOST_CONFIG=host-master.xml
   ```

3. Add the S3 domain controller discovery configuration to the `domain-ec2.xml` file:

   ```xml
   <local>
     <discovery-options>
       <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
            code="org.jboss.as.host.controller.discovery.S3Discovery">
         <property name="access-key" value="S3_ACCESS_KEY"/>
         <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>
         <property name="location" value="S3_BUCKET_NAME"/>
       </discovery-option>
     </discovery-options>
   </local>
   ```
CHAPTER 5. LAUNCHING CLUSTERED JBOSS EAP

5.1. LAUNCH CLUSTERED JBOSS EAP AMIS (WITHOUT MOD_CLUSTER AND VPC)

This topic lists the steps to launch clustered JBoss EAP AMIs without mod_cluster and VPC.

NOTE

- You can use the example configuration scripts that are provided with the image.

See System Paths for more details about system paths.

See Configuring JBoss EAP Subsystems to Work on Cloud Platforms for information about configuring JBoss EAP subsystems for Amazon EC2.

To start clustered JBoss EAP AMI on a standalone server instance, you can use the example /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml file that contains a preconfigured S3_PING JGroups stack. For more information, see S3_PING in the Reliable group communication with JGroups document. This standalone-ec2-ha.xml profile file must be copied from /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/ to the JBoss EAP configuration directory /opt/rh/eap7/root/usr/share/wildfly/standalone/configuration/. Then, you have to add the following line to the JBoss EAP service configuration file:

```
WILDFLY_SERVER_CONFIG=standalone-ec2-ha.xml
```

A unique instance-id needs to be set for each standalone server instance in the undertow subsystem. A value for the instance-id can be set manually by editing the standalone-ec2-ha.xml file or by using the management CLI. For example, you can set the instance-id using the management CLI as follows:

```
/subsystem=undertow:write-attribute(name=instance-id,value=${jboss.jvmRoute})
```

A value for jboss.jvmRoute can then be specified in standalone.conf using the JAVA_OPTS variable.

The jgroups subsystem in the EC2 configuration file requires some S3_PING specific properties to discover cluster members. You must specify access key to S3, secret access key, and the S3 bucket to use for discovery. These properties can either be specified as Java options or put directly into the XML file by editing it or using CLI.

You need to create an S3 bucket for discovery. See Amazon Simple Storage Service Documentation for more information. You may also have to configure the required permissions. The JGroups stack needs to be bound to an IP address, which is used to communicate with other nodes. This can be done by adding Java options, along with S3 Java options to the /opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf file. For example, if the private IP address was 10.10.10.10, then you would add the following line to the standalone.conf file:

```
JAVA_OPTS="-Djboss.bind.address.private=10.10.10.10"
```

You can deploy a sample application: /opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war and observe the logs in /opt/rh/eap7/root/usr/share/wildfly/standalone/log/server.log to see that the JBoss EAP servers have created a cluster.
For Domain Controller Instance

1. Copy the domain-ec2.xml file from 
   /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs to the JBoss EAP configuration
directory.

2. Set the following variables in the appropriate service configuration file:

   WILDFLY_SERVER_CONFIG=domain-ec2.xml
   WILDFLY_HOST_CONFIG=host-master.xml

3. Add S3 domain controller discovery configuration to the host-master.xml file:

   <local>
     <discovery-options>
       <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
 code="org.jboss.as.host.controller.discovery.S3Discovery">
         <property name="access-key" value="S3_ACCESS_KEY"/>
         <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>
         <property name="location" value="S3_BUCKET_NAME"/>
       </discovery-option>
     </discovery-options>
   </local>

4. Configure users and add the secret values for users to the host controller instances. For more
   information, see Create a Managed Domain on Two Machines in the JBoss EAP Configuration
   Guide.

For Host Controller Instance

1. Set the following variable in the appropriate service configuration file:

   WILDFLY_HOST_CONFIG=host-slave.xml

2. Add S3 domain controller discovery configuration to the host-slave.xml file:

   <remote security-realm="ManagementRealm">
     <discovery-options>
       <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
 code="org.jboss.as.host.controller.discovery.S3Discovery">
         <property name="access-key" value="S3_ACCESS_KEY"/>
         <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>
         <property name="location" value="S3_BUCKET_NAME"/>
       </discovery-option>
     </discovery-options>
   </remote>

NOTE

For information about S3 domain controller discovery, see Launch One or More
Instances to Serve as Host Controllers.
WARNING

Running a JBoss EAP cluster in a subnet with network mask smaller than 24-bits or spanning multiple subnets complicates acquiring a unique server peer ID for each cluster member.

IMPORTANT

The auto-scaling Amazon EC2 feature can be used with JBoss EAP cluster nodes. However, ensure it is tested before deployment. You should ensure that your particular workloads scale to the required number of nodes, and that the performance meets your needs for the instance type you are planning to use, different instance types receive a different share of the EC2 cloud resources.

Furthermore, instance locality and current network/storage/host machine/RDS utilization may affect cluster performance. Test with your expected real-life loads and try to account for unexpected conditions.

WARNING

The Amazon EC2 scale-down action terminates the nodes without any chance to gracefully shut down and as some transactions might be interrupted, other cluster nodes and load balancers need time to fail over. This is likely to impact your application users’ experience.

It is recommended that you either scale down the application cluster manually by disabling the server from the mod_cluster management interface until processed sessions are completed, or shut down the JBoss EAP instance gracefully using SSH access to the instance or Red Hat JBoss Operations Network.

Test your procedure for scaling down does not lead to adverse effects on your users’ experience. Additional measures might be required for particular workloads, load balancers, and setups.

5.2. LAUNCH CLUSTERED JBOSS EAP AMIS (WITH MOD_CLUSTER AND VPC)

This topic lists the steps to launch an Apache HTTP server instance to serve as a mod_cluster proxy and a NAT instance for the Virtual Private Cloud (VPC).

See System Paths for more details about system paths.

See Configuring JBoss EAP Subsystems to Work on Cloud Platforms for information about configuring JBoss EAP subsystems for Amazon EC2.
NOTE

- You can use the example configuration scripts that are provided with the image.

An Amazon Virtual Private Cloud (Amazon VPC) is a feature of Amazon Web Services (AWS) that allows you to isolate a set of AWS resources in a private network. The topology and configuration of this private network can be customized to your needs.

See Amazon Virtual Private Cloud for more information about Amazon VPC.

NOTE

If you start a cluster with a mod_cluster load balancer inside a VPC, the JBoss EAP servers are inaccessible to public. The mod_cluster load balancer can be the only endpoint that is connected to the Internet.

See Launch an Instance to Serve as a Domain Controller for setting up domain controller instance.

See Launch One or More Instances to Serve as Host Controllers for setting up host controller instance.

See Launch One or More Instances to Serve as Host Controllers for information about S3 domain controller discovery.

5.2.1. To launch clustered AMIs with VPC and mod_cluster

NOTE

Configuring the VPC is optional. See the Detecting Your Supported Platforms and Whether You Have a Default VPC section in the Amazon VPC user guide for more information.

1. Install jbcs-httpd24-mod_cluster-native package and all of its dependencies. The mod_cluster configuration file is installed in /opt/rh/jbcs-httpd24/root/etc/httpd/conf.d/mod_cluster.conf.

See the Apache HTTP Server Installation Guide for more information about installation of Red Hat JBoss Core Services Apache HTTP Server.

2. Disable advertising for mod_cluster. Add the following to VirtualHost in the /opt/rh/jbcs-httpd24/root/etc/httpd/conf.d/mod_cluster.conf configuration file.

   ```
   ServerAdvertise Off
   EnableMCPMReceive
   # AdvertiseFrequency # comment out AdvertiseFrequency if present
   ```

2. Allow ports in SELinux. If required, configure the iptables. Ports can be allowed in SELinux by using the semanage port -a -t http_port_t -p tcp $PORT_NR command.

3. Configure JBoss EAP to look for mod_cluster proxy on the address that mod_cluster listens on.
NOTE

An /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml example configuration file is provided. You need to configure a list of proxies in the modcluster subsystem.

You can define a list of proxies using one of the following methods:

- Define an outbound-socket-binding called mod-cluster-proxy1 with an appropriate host and port:

  ```xml
  <outbound-socket-binding name="mod-cluster-proxy1">
    <remote-destination host="${jboss.modcluster.proxy1.host}" port="${jboss.modcluster.proxy1.port}"/>
  </outbound-socket-binding>
  ```

- Set the proxies attribute in the modcluster subsystem to mod-cluster-proxy1 with an appropriate host and port:

  ```xml
  /socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=mod-cluster-proxy1:add(host=
  {${jboss.modcluster.proxy1.host}}, port={${jboss.modcluster.proxy1.port}})
  ```
CHAPTER 6. TROUBLESHOOTING

6.1. ABOUT TROUBLESHOOTING AMAZON EC2

EC2 provides an Alarm Status for each instance, indicating severe instance malfunction but the absence of such an alarm is no guarantee that the instance has started correctly and services are running properly. It is possible to use Amazon CloudWatch with its custom metric functionality to monitor instance services' health but use of an enterprise management solution is recommended.

6.2. DIAGNOSTIC INFORMATION

In case of a problem being detected by the JBoss Operations Network, Amazon CloudWatch or manual inspection, common sources of diagnostic information are:

- **/var/log** also contains all the logs collected from machine startup, JBoss EAP, httpd and most other services.

  JBoss EAP log files can be found in `/opt/rh/eap7/root/usr/share/wildfly/`.

Access to these files is only available using an SSH session.

See [Getting Started with Amazon EC2 Linux Instances](#) for more information about how to configure and establish an SSH session with an Amazon EC2 instance.
APPENDIX A. REFERENCE MATERIAL

A.1. AMAZON EC2 AMIS FOR RED HAT CLOUD ACCESS PROGRAM

AMIs are a basic RPM install of JBoss EAP + JDK in the Red Hat Enterprise Linux image, with potentially an Amazon EC2 example configuration. Advanced scripting is no longer available, however regular bash scripts can be used.

AMIs for Platform/JDK Combinations:

- RHEL 6 + Open JDK 8 (1 image)
- RHEL 7 + Open JDK 8 (1 image)

**NOTE**

Both platforms should be of 64-bit architecture.

Maintenance of AMIs

`yum update` should be run regularly, to apply z releases (patches) on EC2. New AMIs for the y releases (minor releases) will be provided by Red Hat.

Scenario 1 (Supported)

1. Sign up for EC2.
2. Sign up for Red Hat Cloud Access.
3. Select the Red Hat AMI from the list of available AMIs.
4. (Optional) Customize JBoss EAP configuration using user scripts or `ssh`.
5. Maintenance: `yum update` for z releases, new AMI for the y releases.

A.2. EXAMPLE CONFIGURATION FILES AND DEPLOYMENTS

The following package adds example deployments for RHEL 7 AMI version:

```
$ rpm -ql eap7-jboss-ec2-eap-samples
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/hello.war
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/jboss-as-helloworld-mdb-7.0.0.ER5-redhat-1.war
```

The JBoss EAP example configuration files contain a JGroups stack set up for the **S3_PING** protocol that can be used for creating clusters across EC2. For the exact location of the example configuration files, see System Paths.

Both the configuration files contain the **modcluster** subsystem to use proxy mod_cluster discovery instead of advertising, because multicast is disabled on EC2.

A.3. SYSTEM PATHS
Service Configuration Files:
- RHEL 6: /etc/sysconfig/*
- RHEL 7: /etc/opt/rh/eap7/wildfly/*

JBoss EAP Home:
- /opt/rh/eap7/root/usr/share/wildfly/

JBoss EAP Configuration Locations:

Standalone instance
- /opt/rh/eap7/root/usr/share/wildfly/standalone/configuration
- /opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf

Managed domain
- /opt/rh/eap7/root/usr/share/wildfly/bin/domain.conf
- /opt/rh/eap7/root/usr/share/wildfly/domain/configuration

JBoss EAP Example Configuration Locations:
- /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml
- /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-full-ha.xml

A.4. LAUNCHING JBOSS EAP ON AMAZON EC2 USING A SCRIPT

The following sample script can be used to start JBoss EAP bound to a public IP address when you launch a JBoss EAP instance on Amazon EC2.

```bash
#!/bin/bash

# platform dependent variables
if [[ "\"cat /etc/redhat-release"" = "\"release 7\"" ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-standalone.conf
    START_COMMAND="systemctl start eap7-standalone"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-standalone
    START_COMMAND="service eap7-standalone start"
fi

# set up addresses
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' | sed "s-/24--g" | cut -d'/' -f1`

JAVA_OPTS="JAVA_OPTS -$Djboss.bind.address=$INTERNAL_IP_ADDRESS -Djboss.bind.address.private=$INTERNAL_IP_ADDRESS -Djboss.bind.address.management=$INTERNAL_IP_ADDRESS" >> /opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf

# start EAP
$START_COMMAND
```
A.5. CONFIGURING JBOSS EAP SUBSYSTEMS TO WORK ON CLOUD PLATFORMS

Some JBoss EAP subsystems must be configured to work properly on cloud platforms, such as Amazon EC2 and Microsoft Azure. This is required because a JBoss EAP server is usually bound to a cloud virtual machine’s private IP address, for example: 10.x.x.x, which is only visible from within the cloud platform. For certain subsystems, this address must also be mapped to a server’s public IP address, which is visible from outside the cloud.

A.5.1. Web Services

When a client makes a web service request using `Service.create(wsdlURL, serviceName);`, the user connects to the server public IP address, but is subsequently redirected to an address defined in the server configuration files in the `webservices` subsystem. By default, this address is `${jboss.bind.address:127.0.0.1}`, which means that on a cloud platform, the caller will be redirected to the server’s private IP address and will be unable to resolve the request. The server’s public IP address has to be configured in the `wsdl-host` element, using the following command:

```
/subsystem=webservices:write-attribute(name=wsdl-host,value=PUBLIC_IP_ADDRESS)
```

A.5.2. Messaging

When using messaging on a cloud platform, the connection factory that the client uses must have a connector pointing to the server’s public IP address.

For this reason a new connector and socket binding must be created for JBoss EAP servers running a full profile.

1. The referenced `http-public` socket binding must be created within the `socket-binding-group`:

```
```

2. Create the new `http-connector` element in the `messaging` subsystem:

```
```

3. Set the `connectors` in the `connection-factory`, which will be used by clients. For example, configuration of `RemoteConnectionFactory` as the default connection will be:

```
/subsystem=messaging-activemq/server=default/connection-factory=RemoteConnectionFactory:write-attribute(name=connectors, value=["http-public-connector"]
```

A.5.3. Remoting Configuration for High Availability

If you are using JBoss EAP HA features with clustered Jakarta Enterprise Beans on a cloud platform, some extra configuration for the `remoting` subsystem is required to ensure Jakarta Enterprise Beans clients can receive cluster view updates.
This is done by configuring `client-mappings` for the `remoting` subsystem socket binding:

```bash
/socket-binding-group=standard-sockets/socket-binding=http:write-attribute(name=client-mappings,value=[{ "destination-address" => "PUBLIC_IP_ADDRESS", "destination-port" => "8080" }])
```

## A.6. Example User Data for Clustered JBoss EAP Instances

The following examples show user data configured for several different server configurations.

### Example: File for Standalone Mode on RHEL6/7

```bash
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a standalone instance using the ec2-ha profile
# This file is for RHEL 6/7, standalone mode only
### This script makes use of the following four Bash variables for clustering setup,
### be sure to add in your own values for these variables here when copy/pasting this
### script into the EC2 user data field

ACCESS_KEY_ID=<your AWS access key>
SECRET_ACCESS_KEY=<your AWS secret access key>
S3_PING_BUCKET=<your bucket name>
NODE_NAME=<your node name>

#### No further modifications should be needed below to run this example ####
# Set the location of JBoss EAP
JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly

# Set the internal IP address of this EC2 instance which is mapped to a public address
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' | sed "s-/24--g" | cut -d'/' -f1`

# Set the location of the standalone.conf file and set the command used to start EAP in standalone mode
if [[ "`cat /etc/redhat-release`" = *"release 7"* ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-standalone.conf
    START_COMMAND="systemctl start eap7-standalone"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-standalone
    START_COMMAND="service eap7-standalone start"
fi

# Configure JBoss EAP to use the ec2-ha profile
cp /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml $JBOSS_HOME/standalone/configuration/standalone-ec2-ha.xml
echo "WILDFLY_SERVER_CONFIG=standalone-ec2-ha.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_BIND=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS=""\"$JAVA_OPTS -Djboss.jgroups.s3_ping.access_key=$ACCESS_KEY_ID -Djboss.jgroups.s3_ping.secret_access_key=$SECRET_ACCESS_KEY -Djboss.jgroups.s3_ping.bucket=$S3_PING_BUCKET -Djboss.jvmRoute=$NODE_NAME\""" >> $JBOSS_HOME/bin/standalone.conf
```

APPENDIX A. REFERENCE MATERIAL
# Deploy the sample application from the local filesystem

```
cp /opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war
$JBOSS_HOME/standalone/deployments/
```

# Start JBoss EAP, note that RHEL 7 does not wait for JBoss EAP to start before returning from the service start. In some cases, there could be a delay of more than 90 seconds.

```
$START_COMMAND
```

Example: File for Starting a Clustered Domain Instance (Domain Controller)

```
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a domain controller with clustering enabled
# This file is for RHEL 6/7, domain controller, domain mode only
### This script makes use of the following Bash variables for clustering and domain controller discovery setup, be sure to add in your own values for these variables here
### when copy/pasting this script into the EC2 user data field

ACCESS_KEY_ID=<your access key id>
SECRET_ACCESS_KEY=<your secret access key>
S3_PING_BUCKET=<your s3 ping bucket>

#### No further modifications should be needed below to run this example ####
# Set the location of JBoss EAP
JBoss_HOME=/opt/rh/eap7/root/usr/share/wildfly
CONF_FILE=/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/domain-ec2.xml

# Set the internal IP address of this EC2 instance which is mapped to a public address
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' | sed "s-/24--g" | cut -d'/" -f1`

# Set the location of the domain.conf file and set the command used to start EAP in domain mode
if [[ "" = "" ]]; then
   SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-domain.conf
   START_COMMAND="systemctl start eap7-domain"
else
   SERVICE_CONF_FILE=/etc/sysconfig/eap7-domain
   START_COMMAND="service eap7-domain start"
fi

# Configure JBoss EAP to use the domain-ec2.xml and host-master.xml configuration files

```
cp $CONF_FILE $JBoss_HOME/domain/configuration/domain-ec2.xml
echo "WILDFLY_SERVER_CONFIG=domain-ec2.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_HOST_CONFIG=host-master.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_BINDING_ADDRESS=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS="" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS=-Djboss.jgroups.s3_ping.access_key="$ACCESS_KEY_ID" -Djboss.jgroups.s3_ping.secret_access_key="$SECRET_ACCESS_KEY" -Djboss.jgroups.s3_ping.bucket="$S3_PING_BUCKET"" >> $JBoss_HOME/bin/domain.conf
```

```
Djboss.bind.address.private=$INTERNAL_IP_ADDRESS -Djboss.bind.address.management=$INTERNAL_IP_ADDRESS" >> $JBOSS_HOME/bin/domain.conf

echo 'HOST_CONTROLLER_JAVA_OPTS="$HOST_CONTROLLER_JAVA_OPTS $JAVA_OPTS"' >> $JBOSS_HOME/bin/domain.conf

# Add a management user with the following credentials:
# User name: admin
# Password: secret_Passw0rd
$JBOSS_HOME/bin/add-user.sh -u admin -p secret_Passw0rd -e -g Management

# Update the main-server-group in domain-ec2.xml to use the ec2-ha profile
$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --domain-config=domain-ec2.xml, /server-group=main-server-group:write-attribute(name=profile, value=ha)"

# Need to modify permissions since this script is executed as the root user
chgrp jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chgrp jboss $JBOSS_HOME/domain/log/audit.log
chgrp jboss $JBOSS_HOME/domain/log/host-controller.log
chown jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chown jboss $JBOSS_HOME/domain/log/audit.log
chown jboss $JBOSS_HOME/domain/log/host-controller.log

# Configure S3 domain controller discovery
yum install patch -y
cd $JBOSS_HOME/domain/configuration
echo "--- host-master.xml 2016-03-18 17:34:26.000000000 -0400
+++ host-master2.xml 2016-04-11 08:28:02.771000191 -0400
@@ -54,7 +54,15 @@
  </management-interfaces>
 </management>
 <domain-controller>
 - <local/>
 +<local>
 + <discovery-options>
 + <discovery-option name="s3-discovery" module=\"org.jboss.as.host-controller\" code=\"org.jboss.as.host.controller.discovery.S3Discovery\">
 + <property name="access-key" value="\$ACCESS_KEY_ID\"/>
 + <property name="secret-access-key" value="\$SECRET_ACCESS_KEY"/>
 + <property name="location" value="\$S3_PING_BUCKET"/>
 + </discovery-option>
 + </discovery-options>
 +</local>
 </domain-controller>
 <interfaces>
  <interface name="management">
    | patch host-master.xml
  
cd -

# Start JBoss EAP, do not forget that RHEL 7 does not wait for JBoss EAP to start before returning
from the service start. In some cases, there could be a delay of more than 90 seconds.

```
$START_COMMAND
sleep 20
# Set up EC2 HA socket bindings for main server group
$JBOSS_HOME/bin/jboss-cli.sh -c --controller=$INTERNAL_IP_ADDRESS:9990 --timeout=120000 -
-command="/server-group=main-server-group:write-attribute(name=socket-binding-group,value=ha-
sockets)"

# Deploy the sample application from the local filesystem to the main-server-group
$JBOSS_HOME/bin/jboss-cli.sh -c --controller=$INTERNAL_IP_ADDRESS:9990 --timeout=120000 -
-command=deploy /opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war --
server-groups=main-server-group"
```

Example: File for Starting a Clustered Domain Instance (Host Controller)

```
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a host
controller with clustering enabled
# This file is for RHEL 6/7, host controller, domain mode only
### This script makes use of the following Bash variables for clustering and domain
### controller discovery setup, be sure to add in your own values for these variables here
### when copy/pasting this script into the EC2 user data field

ACCESS_KEY_ID=<your access key id>
SECRET_ACCESS_KEY=<your secret access key>
S3_PING_BUCKET=<your s3 ping bucket>

#### No further modifications should be needed below to run this example ####

# Set the location of EAP
JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly

# Set the internal IP address of this EC2 instance which is mapped to a public address
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' |
| sed "s-/24--g" | cut -d'/' -f1`

# Set the location of the domain.conf file and set the command used to start EAP in domain mode
if [ ""cat /etc/redhat-release"" = ""release 7"" ]; then
  SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-domain.conf
  START_COMMAND="systemctl start eap7-domain"
else
  SERVICE_CONF_FILE=/etc/sysconfig/eap7-domain
  START_COMMAND="service eap7-domain start"
fi

# Configure variables needed by JBoss EAP
echo "WILDFLY_BIND=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "WILDFLY_HOST_CONFIG=host-slave.xml" >> $SERVICE_CONF_FILE
echo "$JAVA_OPTS="/"$JAVA_OPTS -Djboss.jgroups.s3.ping.access_key="$ACCESS_KEY_ID" -
-Djboss.jgroups.s3.ping.secret_access_key="$SECRET_ACCESS_KEY" -
-Djboss.jgroups.s3.ping.bucket="$S3_PING_BUCKET"" >> $JBOSS_HOME/bin/domain.conf
```

```
$JBOSS_HOME/bin/domain.conf
```

Red Hat JBoss Enterprise Application Platform 7.4 Deploying JBoss EAP on Amazon Web Services

26
echo 'HOST_CONTROLLER_JAVA_OPTS="$HOST_CONTROLLER_JAVA_OPTS $JAVA_OPTS"'
>> $JBOSS_HOME/bin/domain.conf

# Configure S3 domain controller discovery
yum install patch -y
cd $JBOSS_HOME/domain/configuration

echo "--- host-slave.xml.orig 2016-06-07 09:55:27.183390617 +0200
+++ host-slave.xml 2016-06-07 09:56:52.540170784 +0200
@@ -57,7 +57,11 @@
<domain-controller>
    <remote security-realm="ManagementRealm">
        <discovery-options>
            -                <static-discovery name="primary" protocol="\${jboss.domain.master.protocol:remote}" host="\${jboss.domain.master.address}" port="\${jboss.domain.master.port:9990}"/>
+                <discovery-option name="s3-discovery" module="org.jboss.as.host-controller" code="org.jboss.as.host.controller.discovery.S3Discovery">
+                    <property name="access-key" value="$ACCESS_KEY_ID"/>
+                    <property name="secret-access-key" value="$SECRET_ACCESS_KEY"/>
+                    <property name="location" value="$S3_PING_BUCKET"/>
+                </discovery-option>
        </discovery-options>
    </remote>
</domain-controller>
"
 | patch host-slave.xml

sed -i 's/ <!--.*-->//g' host-slave.xml # remove nasty '!' signs which break bash
sed -i '/^[ ]*$/d' host-slave.xml # remove nasty lines with ' ' whitespaces which break the patch

EAP_HOST_NAME=`$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --host-config=host-slave.xml, :read-resource" | grep "host:" | cut -d"" -f4`
$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --host-config=host-slave.xml, /host=$EAP_HOST_NAME/core-service=management/security-realm=ManagementRealm/server-identity=secret:write-attribute(name=value,value=c2VjcmV0X1Bhc3N3MHJk)"

sed -i 's/<host xmlns="urn:jboss:domain:8.0">/<host xmlns="urn:jboss:domain:8.0" name="admin">/' host-slave.xml
sed -i 's/other-server-group/main-server-group/' host-slave.xml

cd -

# Need to modify permissions since this script is executed as the root user
chgrp jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chgrp jboss $JBOSS_HOME/domain/log/audit.log
chgrp jboss $JBOSS_HOME/domain/log/host-controller.log
chown jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chown jboss $JBOSS_HOME/domain/log/audit.log
chown jboss $JBOSS_HOME/domain/log/host-controller.log

# Start JBoss EAP, do not forget that RHEL 7 does not wait for JBoss EAP to start before returning from the service start. In some cases, there could be a delay of more than 90 seconds.
$START_COMMAND
Revised on 2022-06-27 13:35:01 UTC