



Red Hat JBoss Enterprise Application Platform 8.0

Deploying JBoss EAP on Amazon Web Services

For Use with Red Hat JBoss Enterprise Application Platform 8.0

Red Hat JBoss Enterprise Application Platform 8.0 Deploying JBoss EAP on Amazon Web Services

For Use with Red Hat JBoss Enterprise Application Platform 8.0

Legal Notice

Copyright © 2024 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 provides information about deploying Red Hat JBoss Enterprise Application Platform on Amazon EC2.

Table of Contents

PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. ABOUT RED HAT CLOUD ACCESS	5
CHAPTER 2. ABOUT AMAZON EC2	6
2.1. TYPES OF JBOSS EAP AMAZON MACHINE IMAGES	6
2.2. RED HAT CLOUD ACCESS FEATURES	6
2.3. SUPPORTED AMAZON EC2 INSTANCE TYPES	7
2.4. SUPPORTED RED HAT AMIS	7
CHAPTER 3. LAUNCHING A JBOSS EAP INSTANCE	8
3.1. LAUNCHING A JBOSS EAP INSTANCE FROM THE AWS MARKETPLACE	8
3.2. LAUNCHING JBOSS EAP INSTANCE FROM PRIVATE AMI USING AWS EC2 CONSOLE	8
CHAPTER 4. LAUNCHING A NON-CLUSTERED JBOSS EAP INSTANCE	10
CHAPTER 5. LAUNCHING NON-CLUSTERED MANAGED DOMAIN	12
5.1. LAUNCHING A NON-CLUSTERED INSTANCE TO SERVE AS A DOMAIN CONTROLLER	12
5.2. LAUNCH ONE OR MORE INSTANCES TO SERVE AS HOST CONTROLLERS	13
CHAPTER 6. LAUNCHING CLUSTERED JBOSS EAP	15
6.1. LAUNCH CLUSTERED JBOSS EAP AMIS WITHOUT MOD_CLUSTER AND VPC	15
6.1.1. Launching clustered AMIs without mod_cluster and VPC for domain controller instance	15
6.1.2. Launching clustered AMIs without mod_cluster and VPC for host controller	16
6.2. LAUNCH CLUSTERED JBOSS EAP AMIS WITH MOD_CLUSTER AND VPC	17
CHAPTER 7. TROUBLESHOOTING	20
7.1. ABOUT TROUBLESHOOTING AMAZON EC2	20
7.2. DIAGNOSTIC INFORMATION	20

PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

Procedure

1. Click the following link to [create a ticket](#).
2. Enter a brief description of the issue in the **Summary**.
3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.
4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

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 RED HAT CLOUD ACCESS

If you have an existing Red Hat subscription, Red Hat Cloud Access 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 cost-effectively move your subscriptions between traditional servers and public cloud-based resources.

You can find more information about [Red Hat Cloud Access on the Customer Portal](#) .

CHAPTER 2. 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.

2.1. TYPES OF JBOSS EAP AMAZON MACHINE IMAGES

Use JBoss EAP on Amazon Elastic Compute Cloud (Amazon EC2) by deploying a public or private Amazon Machine Image (AMI).



IMPORTANT

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

JBoss EAP public AMI

Access JBoss EAP public AMIs through the [AWS marketplace](#). The public AMIs are offered with the pay-as-you-go (PAYG) model. With a PAYG model, you only pay based on the number of computing resources you used.

JBoss EAP private AMI

You can use your existing subscription to access JBoss EAP private AMIs through Red Hat Cloud Access. For information about Red Hat Cloud Access, see [About Red Hat Cloud Access](#).

2.2. 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.

2.3. 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.

The minimum virtual hardware requirements for an AMI to deploy JBoss EAP are the following:

- Virtual CPU: 2
- Memory: 4 GB

However, depending on the applications you deploy on JBoss EAP you might require additional processors and memory.

2.4. SUPPORTED RED HAT AMIS

The supported Red Hat AMIs can be identified by their names, as shown in the following examples:

Private image example

```
RHEL-9-JBEAP-8.0.0_HVM_GA-20240909-x86_64-0-Access2-GP2
```

Public image example

```
RHEL-9-JBEAP-8.0.0_HVM_GA-20240804-x86_64-0-Marketplace-GP2
```

- **RHEL-x** is the version number of Red Hat Enterprise Linux installed in the AMI. Example **9**.
- **JBEAP-x.y.z** is the version number of JBoss EAP installed in the AMI. Example **8.0.0**.
- **20240804** is the date that the AMI was created in the format of YYYYMMDD.
- **x86_64** is the architecture of the AMI. This can be **x86_64** or **i386**.
- **Access2** or **Marketplace** denote whether the AMI is private or public as follows:
 - Private image contains **Access2**.
 - Public image contains **Marketplace**.

CHAPTER 3. LAUNCHING A JBOSS EAP INSTANCE

The following procedures show launching a public JBoss EAP instance from the Amazon Web Services (AWS) marketplace and launching a JBoss EAP instance on Amazon EC2 Console.

3.1. LAUNCHING A JBOSS EAP INSTANCE FROM THE AWS MARKETPLACE

The public JBoss EAP Amazon Machine Image (AMI), offered with the pay-as-you-go (PAYG) model, is available at the Amazon Web Services (AWS) marketplace.

Prerequisite

- You have an AWS account.
- The Amazon Web Services CLI is installed and configured with your account credentials.

Procedure

1. Go to AWS marketplace at the URL: <https://aws.amazon.com/marketplace>.
2. Search for "JBoss EAP" in the search bar. Filter the results by **Publisher**, selecting **Red Hat Limited** and **Red Hat**.
3. Click the image you want to launch.



NOTE

If you are based in Europe, the Middle East, or Africa, select the image from the publisher "**Red Hat Limited**", otherwise select the image from the publisher "**Red Hat**".

You are redirected to the software subscription page.

4. Select the subscription settings and click **Continue to Subscribe**.
5. Accept the terms by clicking **Accept Terms**, click **Continue to Configuration**.
You are redirected to the configuration page.
6. Select the configuration options and click **Continue to Launch**.
You are directed to launch the software page.
7. Review the launch configuration details and launch the instance by clicking **Launch**.

3.2. LAUNCHING JBOSS EAP INSTANCE FROM PRIVATE AMI USING AWS 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.

Prerequisites

- You have a Red Hat subscription.
- You have an AWS account.
- The Amazon Web Services CLI is installed and configured with your account credentials.

Procedure

1. Open the [Amazon EC2 console](#).
2. From the Amazon EC2 console, click **AMIs**.
3. Search for **jbeap** AMI in **Private images**, located in the Amazon Machine Images (AMIs) panel, and select the AMI. For example, **RHEL-9-JBEAP-8.0.0_HVM_GA-20240909-x86_64-0-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.



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 4. 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) created through a private AMI or public Marketplace listing.

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.



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 -
```

Procedure

1. Launch the Red Hat AMI instance.

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



NOTE

- For complex configuration, you can either use the **standalone.conf** file in the JBoss EAP **bin** directory: `/opt/rh/eap8/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.

1. Start JBoss EAP using the following command:

```
$ systemctl start eap8-standalone
```

2. Stop JBoss EAP using the following command:

```
$ systemctl stop eap8-standalone
```



NOTE

If you want to bind JBoss EAP to a different IP address, add the following line in the **/etc/opt/rh/eap8/wildfly/eap8-standalone.conf** file on RHEL 9. The internal IP address is translated into a public IP address by EC2.

```
WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS
```

CHAPTER 5. LAUNCHING NON-CLUSTERED MANAGED DOMAIN

5.1. LAUNCHING A NON-CLUSTERED 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) created through a private AMI or public Marketplace listing.

Prerequisite

- A suitable Red Hat AMI. See [Supported Red Hat AMIs](#) for more information.



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 -
```

Procedure

1. Launch the Red Hat AMI instance.

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



NOTE

- For complex configuration, you can either use the **domain.conf** file in the JBoss EAP **bin** directory: **/opt/rh/eap8/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.

1. Start JBoss EAP using the following command:

```
$ systemctl start eap8-domain
```

2. Stop JBoss EAP using the following command:

```
$ systemctl stop eap8-domain
```


**NOTE**

If you want to bind JBoss EAP to a different IP address, add the following line in the `/etc/opt/rh/eap8/wildfly/eap8-domain.conf` file on RHEL 9. The internal IP address is translated into a public IP address by EC2.

```
WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS
```

5.2. 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](#).

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.

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/eap8/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:

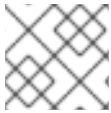
```
<local>
  <discovery-options>
    <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
```

```
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 6. LAUNCHING CLUSTERED JBOSS EAP

6.1. 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.

To start clustered JBoss EAP AMI on a standalone server instance, you can use the example `/opt/rh/eap8/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/eap8/root/usr/share/wildfly/docs/examples/configs/` to the JBoss EAP configuration directory `/opt/rh/eap8/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/eap8/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="$JAVA_OPTS -Djboss.bind.address.private=10.10.10.10
-Djboss.jgroups.aws.s3_ping.region_name=<S3_REGION_NAME>
-Djboss.jgroups.aws.s3_ping.bucket_name=<S3_BUCKET_NAME>"
```

You can deploy a sample application: `/opt/rh/eap8/root/usr/share/java/eap8-jboss-ec2-eap-samples/cluster-demo.war` and observe the logs in `/opt/rh/eap8/root/usr/share/wildfly/standalone/log/server.log` to see that the JBoss EAP servers have created a cluster.

6.1.1. Launching clustered AMIs without `mod_cluster` and VPC for domain controller instance

Procedure

1. Copy the **domain-ec2.xml** file from `/opt/rh/eap8/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*.

6.1.2. Launching clustered AMIs without mod_cluster and VPC for host controller

Procedure

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.

6.2. LAUNCH CLUSTERED JBOSS EAP AMIS WITH MOD_CLUSTER AND VPC

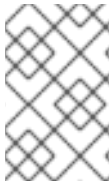
This topic lists the steps to launch an Apache HTTP server instance to serve as a **mod_cluster** proxy and a NAT instance for the Virtual Private Cloud (VPC).

**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.



TO LAUNCH CLUSTERED AMIS WITH VPC AND MOD_CLUSTER

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
```

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



NOTE

An `/opt/rh/eap8/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:

```
<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:

```
/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 7. TROUBLESHOOTING

7.1. 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.

7.2. 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/eap8/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.

Revised on 2024-05-10 16:25:21 UTC