



Red Hat JBoss Web Server 3

Installation Guide

Install and Configure Red Hat JBoss Web Server 3.

Red Hat JBoss Web Server 3 Installation Guide

Install and Configure Red Hat JBoss Web Server 3.

Legal Notice

Copyright © 2016 Red Hat, Inc.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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 book contains information related to installation and basic configuration of Red Hat JBoss Web Server components.

Table of Contents

CHAPTER 1. INTRODUCTION	3
1.1. INTRODUCTION	3
1.2. COMPONENTS	3
1.3. SUPPORTED OPERATING SYSTEMS AND CONFIGURATIONS	4
1.4. INSTALLATION METHODS	4
1.5. UPGRADING JBOSS WEB SERVER	4
1.6. MIGRATING FROM APACHE HTTP SERVER 2.2 TO APACHE HTTP SERVER 2.4	4
1.7. COMPONENT DOCUMENTATION BUNDLE	5
CHAPTER 2. INSTALLING JBOSS WEB SERVER ON RED HAT ENTERPRISE LINUX	6
2.1. PREREQUISITES	6
2.2. ZIP INSTALLATION	8
2.3. RPM INSTALLATION	13
2.4. SELINUX POLICIES	18
CHAPTER 3. INSTALLING JBOSS WEB SERVER ON MICROSOFT WINDOWS	20
3.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)	20
3.2. DOWNLOAD AND EXTRACT JBOSS WEB SERVER	20
3.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION	21
3.4. STARTING JBOSS WEB SERVER	25
3.5. STOPPING JBOSS WEB SERVER	26
CHAPTER 4. INSTALLING JBOSS WEB SERVER ON SOLARIS	27
4.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)	27
4.2. DOWNLOAD AND EXTRACT JBOSS WEB SERVER	27
4.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION	28
4.4. STARTING JBOSS WEB SERVER	30
4.5. STOPPING JBOSS WEB SERVER	31
CHAPTER 5. USING JSVC TO START TOMCAT	32
5.1. STARTING TOMCAT USING JSVC	32
5.2. STOPPING TOMCAT USING JSVC	32
5.3. JSVC PARAMETERS	33
CHAPTER 6. HIBERNATE ON JBOSS WEB SERVER	34
CHAPTER 7. MONITORING JBOSS WEB SERVER WITH JBOSS OPERATIONS NETWORK (ON)	36
7.1. DOWNLOAD THE WEB SERVER PLUGIN PACK FOR JBOSS ON	36
7.2. CONFIGURING APACHE HTTP SERVER FOR JBOSS ON MONITORING	36
7.3. CONFIGURING TOMCAT FOR JBOSS ON MONITORING	37
APPENDIX A. SUPPORTED APACHE HTTP SERVER MODULES	42
APPENDIX B. REVISION HISTORY	43

CHAPTER 1. INTRODUCTION

1.1. INTRODUCTION

Red Hat JBoss Web Server is a fully integrated and certified set of components for hosting Java web applications. It is comprised of a web server (Apache HTTP Server), application server (Apache Tomcat Servlet container), load balancers (mod_jk and mod_cluster), and the Tomcat Native Library.

This Installation Guide includes procedures for the installation, minor upgrade, and basic configuration of JBoss Web Server on supported operating systems.

[Report a bug](#)

1.2. COMPONENTS

Red Hat JBoss Web Server consists of the following components:

- **Apache Tomcat** is a servlet container in accordance with Java Servlet Specification. JBoss Web Server contains Apache Tomcat 7 and Apache Tomcat 8.
- **Apache Native** is a Tomcat library, which improves Tomcat scalability, performance, and integration with native server technologies. It also contains an implementation of APR (Apache Portable Runtime). APR is a binary protocol for communication of Apache HTTP Server and Apache Tomcat.
- **Apache Tomcat Connectors** (mod_jk, mod_cluster) are connectors between Apache HTTP Server and Apache Tomcat. Note that the default is mod_cluster, as it is the JBoss native load balancer and is more efficient, reliable, and scalable than mod_jk.
- **Apache HTTP Server** is an open source web server developed by the [Apache Software Foundation](#). The implementation follows the current HTTP standards.



NOTE

By default, the delivered Apache HTTP Server provides support for Secure Socket Layer (SSL). To remove it, see the procedure on removing SSL in the *Configuring the JBoss Web Server Installation* section for your installation platform.

- **Hibernate** is an object-relational mapping framework. The delivered Hibernate contains Hibernate Core, Hibernate Annotations, Hibernate EntityManager with JPA 1.0 APIs.

For a detailed list of component versions included in JBoss Web Server 3, see <https://access.redhat.com/articles/111723>.



NOTE

Tomcat clustering has been removed from JBoss Web Server 3. If you need clustering or session replication support for Java applications, Red Hat recommends that you use Red Hat JBoss Enterprise Application Platform (JBoss EAP).

[Report a bug](#)

1.3. SUPPORTED OPERATING SYSTEMS AND CONFIGURATIONS

For information on supported operating systems and configurations for Red Hat JBoss Web Server, see <https://access.redhat.com/articles/1377603>.

[Report a bug](#)

1.4. INSTALLATION METHODS

Red Hat JBoss Web Server can be installed on supported Red Hat Enterprise Linux, Microsoft Windows, and Solaris systems using ZIP installation files available for each platform. Red Hat JBoss Web Server can also be installed on supported Red Hat Enterprise Linux systems using RPM packages.

For ZIP installations, below is a summary of what components are included in the two ZIP files that form the core part of a Red Hat JBoss Web Server installation.

- **jws-application-servers-3.0.0-<platform>-<architecture>.zip**
 - Tomcat 7
 - Tomcat 8
 - Platform-specific utilities
- **jws-httpd-3.0.0-<platform>-<architecture>.zip**
 - Apache HTTP Server
 - Apache HTTP Server modules

[Report a bug](#)

1.5. UPGRADING JBOSS WEB SERVER

Upgrading JBoss Web Server is not supported. To upgrade to JBoss Web Server 3 from an older version, a fresh installation is recommended.



IMPORTANT

If the RPM method was used to install JBoss Web Server 1.x or 2.x, it is not possible to remove JBoss Web Server 1.x or 2.x and install JBoss Web Server 3 on the same machine. Many configuration files from the old packages will be left on the system, and will create compatibility issues with the newer RPM packages. For the same reason, manually removing the old RPMs and installing the new ones is also not supported.

On a system where JBoss Web Server 1.x or 2.x was installed using the ZIP method, it is possible to remove the old version and install version JBoss Web Server 3 on the same system.

[Report a bug](#)

1.6. MIGRATING FROM APACHE HTTP SERVER 2.2 TO APACHE HTTP SERVER 2.4

Apache HTTP Server 2.4 is included in Red Hat JBoss Web Server 3. If you are migrating from an environment using Apache HTTP Server 2.2, refer to the Apache Foundation's upgrade advice: <http://httpd.apache.org/docs/2.4/upgrading.html>. An overview of new features in Apache HTTP Server 2.4 is also available: http://httpd.apache.org/docs/2.4/new_features_2_4.html.

[Report a bug](#)

1.7. COMPONENT DOCUMENTATION BUNDLE

Red Hat JBoss Web Server includes an additional documentation bundle that includes the original vendor documentation for each component. This documentation bundle, `jws-docs-3.0.0.zip`, is available at the Red Hat Customer Portal, and contains additional documentation for the following:

- `httpd`
- `mod_auth_kerb`
- `mod_cluster`
- `mod_jk`
- `openssl`
- `tomcat7`
- `tomcat8`
- `tomcat-native`
- `mod_ssl` documentation is also available at the location `doc/httpd/mod/mod_ssl.html`.

[Report a bug](#)

CHAPTER 2. INSTALLING JBOSS WEB SERVER ON RED HAT ENTERPRISE LINUX

You can install Red Hat JBoss Web Server on Red Hat Enterprise Linux using one of two methods:

- ZIP files
- RPM packages

Regardless of which method you choose, you must first install a supported Java Development Kit (JDK).

[Report a bug](#)

2.1. PREREQUISITES

2.1.1. Installing a Java Development Kit (JDK)

Before installing Red Hat JBoss Web Server, you must first install a supported Java Development Kit (JDK).

Procedure 2.1. Installing a Java Development Kit (JDK)

1. Subscribe your Red Hat Enterprise Linux system to the appropriate supplementary channel:
 - Red Hat Enterprise Linux Server Supplementary for Red Hat Enterprise Linux 6
 - Red Hat Enterprise Linux Server Supplementary for Red Hat Enterprise Linux 7
2. As the root user, execute the command to install a 1.7 or 1.8 JDK:

- ```
yum install java-1.7.0-<VENDOR>-devel
```

Replace **<VENDOR>** with **oracle**, **ibm**, or **openjdk**.

- ```
# yum install java-1.8.0-<VENDOR>-devel
```

Replace **<VENDOR>** with **oracle**, **ibm**, or **openjdk**.

3. Run the following commands as the root user to ensure the correct JDK is in use:

```
# alternatives --config java
```

```
# alternatives --config javac
```

These commands return lists of available JDK versions with the selected version marked with a plus (+) sign. If the selected JDK is not the desired one, change to the desired JDK as instructed in the shell prompt.



IMPORTANT

All software that use the `java` and `javac` commands uses the JDK set by `alternatives`. Changing Java alternatives may impact on the running of other software.

[Report a bug](#)

2.1.2. Red Hat Enterprise Linux Package Prerequisites

Before installing Red Hat JBoss Web Server on Red Hat Enterprise Linux, ensure the following prerequisites are met.

- A supported version of Java is installed. See [Section 2.1.1, “Installing a Java Development Kit \(JDK\)”](#).
- The following required packages are installed for Red Hat Enterprise Linux 6:
 - `krb5-workstation`
 - `elinks`
 - `mailcap`. The `mailcap` package contains `/etc/mime.types`, which is used by `httpd` for MIME type mapping extensions. This package is not included in Red Hat Enterprise Linux and must be included.
- The following required packages are installed for Red Hat Enterprise Linux 7:
 - `apr`
 - `apr-devel`
 - `apr-util`
 - `apr-util-devel`
 - `apr-util-ldap`
 - `elinks`
 - `krb5-workstation`
 - `mailcap`
- You must remove the `tomcatjss` package before installing the `tomcat-native` package. The `tomcatjss` package uses an underlying NSS security model rather than the OpenSSL security model.

Procedure 2.2. Removing the `tomcatjss` Package

- As the root user, run the following command to remove `tomcatjss`:

```
# yum remove tomcatjss
```

Procedure 2.3. Installing Required Packages

- For your version of Red Hat Enterprise Linux, run the install command below as the root user:

- For Red Hat Enterprise Linux 6:

```
# yum install krb5-workstation elinks mailcap
```

- For Red Hat Enterprise Linux 7:

```
# yum install apr apr-devel apr-util apr-util-devel apr-util-ldap  
elinks krb5-workstation mailcap
```

[Report a bug](#)

2.2. ZIP INSTALLATION

Ensure that all the required prerequisite packages are installed before installing Red Hat JBoss Web Server. See [Section 2.1.2, “Red Hat Enterprise Linux Package Prerequisites”](#).

[Report a bug](#)

2.2.1. Download and Extract JBoss Web Server

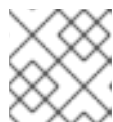
To install Red Hat JBoss Web Server, download and extract the installation ZIP files.

Procedure 2.4. Downloading JBoss Web Server

1. Open a browser and log in to the Customer Portal at <http://access.redhat.com>.
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:
 - Red Hat JBoss Web Server 3.0 Application Server (**jws-application-servers-3.0.0-<platform>-<architecture>.zip**)
 - Red Hat JBoss Web Server 3.0 Apache HTTP Server (**jws-httpd-3.0.0-<platform>-<architecture>.zip**)

Procedure 2.5. Extract JBoss Web Server

- Unzip the downloaded ZIP files to your installation directory.



NOTE

We recommend that you install JBoss Web Server in the `/opt/` directory.

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as `JWS_HOME`.

[Report a bug](#)

2.2.2. Configuring the JBoss Web Server Installation

Some configuration is required before running Red Hat JBoss Web Server. This section includes the following configuration procedures:

- [Procedure 2.6, “Setting the JAVA_HOME Environment Variable”](#)
- Creating Tomcat and Apache users for simple and secure user management: [Procedure 2.7, “Creating a Tomcat User”](#) and [Procedure 2.8, “Creating an Apache User”](#).
- [Procedure 2.9, “Removing/Re-Adding SSL Support”](#)
- [Procedure 2.10, “Enabling log4j Logging for Tomcat”](#)
- [Procedure 2.11, “Configuring mod_jk”](#)
- [Procedure 2.12, “Configuring mod_cluster”](#)
- [Procedure 2.13, “Running the Apache HTTP Server Post-Installation Script”](#)

Procedure 2.6. Setting the JAVA_HOME Environment Variable

You must set the `JAVA_HOME` environment variable for Tomcat before running JBoss Web Server.

- In the `bin` directory of Tomcat (either `JWS_HOME/tomcat7/bin` or `JWS_HOME/tomcat8/bin`), create a file named `setenv.sh`, and insert the `JAVA_HOME` path definition.

For example: `export JAVA_HOME=/usr/lib/jvm/jre-1.7.0-openjdk.x86_64`

Procedure 2.7. Creating a Tomcat User

Follow this procedure to create the `tomcat` user and its parent group:

1. In a shell prompt as the root user, change directory to `JWS_HOME`.
2. Run the following command to create the `tomcat` user group:

```
# groupadd -g 91 -r tomcat
```

3. Run the following command to create the `tomcat` user in the `tomcat` user group:

```
# useradd -c "Tomcat" -u 91 -g tomcat -s /bin/sh -r tomcat
```

4. From `JWS_HOME`, run the following command to assign the ownership of the Tomcat directories to the `tomcat` user to allow the user to run the Tomcat service:

```
# chown -R tomcat:tomcat tomcat<VERSION>
```

Replace `<VERSION>` with the respective Tomcat version number (`7` or `8`).

You can use `ls -l` to verify that the `tomcat` user is the owner of the directory.

5. Ensure that the `tomcat` user has execute permissions to all parent directories. For example:

```
# chmod -R u+X tomcat<VERSION>
```

Procedure 2.8. Creating an Apache User

Follow this procedure to create the `apache` user and its parent group:

1. In a shell prompt as the root user, change directory to `JWS_HOME`.
2. Run the following command to create the `apache` user group:

```
# groupadd -g 48 -r apache
```

3. Run the following command to create the `apache` user in the `apache` user group:

```
# useradd -c "Apache" -u 48 -g apache -s /bin/sh -r apache
```

4. From `JWS_HOME`, run the following command to assign the ownership of the Apache directories to the `apache` user to allow the user to run `httpd`:

```
# chown -R apache:apache httpd
```

You can use `ls -l` to verify that the `apache` user is the owner of the directory.

Procedure 2.9. Removing/Re-Adding SSL Support

JBoss Web Server supports SSL by default, but it can be removed. Follow this procedure to remove or re-add SSL support.

- Go to the `JWS_HOME/httpd/conf.d/` directory and rename the SSL configuration file:
 - To remove SSL, rename `ssl.conf` to `ssl.conf.disabled`.
 - To re-add SSL, rename `ssl.conf.disabled` to `ssl.conf`.

Procedure 2.10. Enabling log4j Logging for Tomcat

If required, follow this procedure to add log4j logging to Tomcat.

1. Open a shell prompt and change directory to `JWS_HOME/extras/`.
2. Copy the `log4j-eap6.jar`, `log4j.properties`, and `tomcat-juli-adapters.jar` files to the `lib` directory of the Tomcat directory.

For example:

```
# cp log4j.properties ../tomcat<VERSION>/lib/
```

```
# cp log4j-eap6.jar ../tomcat<VERSION>/lib/
```

```
# cp tomcat-juli-adapters.jar ../tomcat<VERSION>/lib/
```

-
- Replace `<VERSION>` with the respective Tomcat version number (7 or 8).
- 3. Replace `tomcat-juli.jar` file in your Tomcat `bin` directory with the `tomcat-juli.jar` file from `JWS_HOME/extras/`:

```
# cp tomcat-juli.jar ../tomcat<VERSION>/bin/
```

Procedure 2.11. Configuring mod_jk

If required, you can configure Apache HTTP Server to use `mod_jk` as its load balancer.

The steps below list the locations of the `mod_jk` configuration files. For detailed `mod_jk` configuration instructions, see the *Configure Load Balancing Using Apache HTTP Server and mod_jk* section of the *HTTP Connectors and Load Balancing Guide*.



NOTE

Sample configuration files are provided for `mod_jk`, and are located in `JWS_HOME/httpd/conf.d/`. The sample configuration files are: `mod_jk.conf.sample`, `workers.properties.sample`, and `uriworkermap.properties.sample`. To use these samples instead of creating your own configuration files, remove the `.sample` extension, and modify their content as needed.

1. In `JWS_HOME/httpd/conf.d/`, create a file named `workers.properties`.

This file should contain the available workers (JBoss instances).

2. In `JWS_HOME/httpd/conf.d/`, create a file named `mod_jk.conf`.

This file contains general `mod_jk` configuration.

3. In `JWS_HOME/httpd/conf.d/`, create a file named `uriworkermap.properties`.

This file contains request mapping rules that map requests to workers.

Procedure 2.12. Configuring mod_cluster

- `mod_cluster` can be configured in `JWS_HOME/httpd/conf.d/mod_cluster.conf`.

For more information about customizing `mod_cluster`, see the *mod_cluster Connector* section of the *HTTP Connectors and Load Balancing Guide*

Procedure 2.13. Running the Apache HTTP Server Post-Installation Script

1. In a shell prompt as the root user, change directory to `JWS_HOME/httpd`.
2. Run the following command:

```
# ./postinstall
```

[Report a bug](#)

2.2.3. Starting JBoss Web Server

To start Red Hat JBoss Web Server, you must start the following:

- Tomcat (7 or 8).
- Apache HTTP Server.

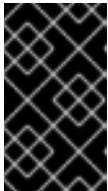
Before starting Tomcat, ensure that the following prerequisites are met:

- The Tomcat user is created. See [Procedure 2.7, “Creating a Tomcat User”](#).
- JAVA_HOME is set correctly. See [Procedure 2.6, “Setting the JAVA_HOME Environment Variable”](#).

Procedure 2.14. Starting Tomcat

- Run the following command as the `tomcat` user with your respective Tomcat version (7 or 8):

```
$ sh JWS_HOME/tomcat<VERSION>/bin/startup.sh
```



IMPORTANT

Although there are multiple methods of starting Tomcat, it is recommended that you use the `startup.sh` script. To start Tomcat as a service using `Jsvc`, see [Chapter 5, Using Jsvc to Start Tomcat](#).

Procedure 2.15. Starting Apache HTTP Server

- To start Apache HTTP Server (`httpd`), in a terminal as the root, change to `JWS_HOME/httpd/sbin/` and run the following command:

```
# ./apachectl start
```

[Report a bug](#)

2.2.4. Stopping JBoss Web Server

To stop Red Hat JBoss Web Server, you must stop the following:

- Tomcat (7 or 8).
- Apache HTTP Server.

Procedure 2.16. Stopping Tomcat

- To stop Tomcat, run the following command as the root user with your respective Tomcat version (7 or 8):

```
# sh JWS_HOME/tomcat<VERSION>/bin/shutdown.sh
```

Procedure 2.17. Stopping Apache HTTP Server

- To stop Apache HTTP Server (httpd), in a shell prompt as the root user change to `JWS_HOME/ht tpd/sbin/`, and run the following command:

```
# ./apachectl stop
```

[Report a bug](#)

2.3. RPM INSTALLATION

Installing JBoss Web Server from RPM packages installs Tomcat and Apache HTTP Server as services, and installs its resources into absolute paths. The RPM installation option is only available for Red Hat Enterprise Linux 6 and Red Hat Enterprise Linux 7.

RPM installation packages for JBoss Web Server are available from Red Hat Subscription Management.

NOTE

For users wanting to manage JBoss Web Server installations using Red Hat Satellite: although Red Hat Satellite 6 is recommended for managing JBoss Web Server 3 installations, the following Red Hat Network (RHN) channels are also provided specifically for Satellite 5 users:

- For Red Hat Enterprise Linux 6:
 - `jws-3-x86_64-server-6-rpm`
 - `jws-3-i386-server-6-rpm`
- For Red Hat Enterprise Linux 7:
 - `jws-3-x86_64-server-7-rpm`

Red Hat Satellite 6 users can use the repositories shown in [Section 2.3.1, “Installing JBoss Web Server from RPM Packages”](#).



WARNING

Installing JBoss Web Server from RPM packages shares Java library files with other applications. Library version conflicts occur when using RPM packages to install both JBoss Web Server 3 and JBoss EAP 6 on the same machine. To work around the issue, you can install either JBoss Web Server 3 or JBoss EAP 6 using the RPM installation method, and the other using the ZIP installation method.

[Report a bug](#)

2.3.1. Installing JBoss Web Server from RPM Packages

Prerequisites

- [Section 2.1.1, “Installing a Java Development Kit \(JDK\)”](#)
- Ensure that the tomcatjss package is removed. See [Section 2.1.2, “Red Hat Enterprise Linux Package Prerequisites”](#)

Before downloading and installing the RPM packages, you must register your system with Red Hat Subscription Management and subscribe to the respective Content Delivery Network (CDN) repositories.

Procedure 2.18. Installing JBoss Web Server from RPM Packages

1. Log in to [Red Hat Subscription Manager](#).
2. Click on **Systems** in the **Subscriber Inventory**.
3. Subscribe to the JBoss Web Server CDN repositories for your operating system version and architecture:
 - For Red Hat Enterprise Linux 6:
 - `jws-3-for-rhel-6-server-rpms`
 - For Red Hat Enterprise Linux 7:
 - `jws-3-for-rhel-7-server-rpms`
4. Run the following command as the root user to install JBoss Web Server:

```
# yum groupinstall jws3
```



NOTE

Although not recommended, instead of using the group install, you can install each of the packages and their dependencies individually.

[Report a bug](#)

2.3.2. Installing the JBoss Web Server Plus Group

The JBoss Web Server Plus group contains additional packages mainly for the addition of Hibernate and its dependencies.

Procedure 2.19. Installing the Red Hat JBoss Web Server Plus Group of Packages

- To install the JBoss Web Server Plus group of packages, run the following command as the root user:

```
# yum groupinstall jws3plus
```

[Report a bug](#)

2.3.3. Configuring the JBoss Web Server Installation (RPM Installation)

Before starting an RPM installation of Red Hat JBoss Web Server, there are some optional configurations you can perform. This section includes the following configuration procedures:

- [Procedure 2.20, “Removing SSL Support”](#)
- [Procedure 2.21, “Configuring mod_jk”](#)
- [Procedure 2.22, “Configuring Apache HTTP Server to Use mod_cluster”](#)

Procedure 2.20. Removing SSL Support

JBoss Web Server supports SSL by default, but it can be removed. To remove SSL support, remove the `mod_ssl` package.

- At a shell prompt, run the following command as the root user:

```
# yum remove mod_ssl24
```

Procedure 2.21. Configuring mod_jk

If required, you can configure Apache HTTP Server to use `mod_jk` as its load balancer.

The steps below list the locations of the `mod_jk` configuration files. For detailed `mod_jk` configuration instructions, see the *Configure Load Balancing Using Apache HTTP Server and mod_jk* section of the *HTTP Connectors and Load Balancing Guide*.



NOTE

Sample configuration files are provided for `mod_jk`, and are located in `/etc/httpd24/conf.d/`. The sample configuration files are: `mod_jk.conf.sample`, `workers.properties.sample`, and `uriworkermap.properties.sample`. To use these samples instead of creating your own configuration files, remove the `.sample` extension, and modify their content as needed.

1. In `/etc/httpd24/conf.d/`, create a file named `workers.properties`.

This file should contain the available workers (JBoss instances).

2. In `/etc/httpd24/conf.d/`, create a file named `mod_jk.conf`.

This file contains general `mod_jk` configuration.

3. In `/etc/httpd24/conf.d/`, create a file named `uriworkermap.properties`.

This file contains request mapping rules that map requests to workers.

Procedure 2.22. Configuring Apache HTTP Server to Use mod_cluster

The `mod_cluster` load balancer is optional. Follow this procedure to configure Apache HTTP Server to use `mod_cluster` as its load balancer.

1. In `/etc/httpd24/conf.modules.d/00-proxy.conf`, ensure that there is a `#` at the beginning of the following line to disable `mod_proxy_balancer.so`:

```
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

This module is incompatible with `mod_cluster`.

2. Configure the server to load the `mod_cluster` modules:
 - a. Create the file `/etc/httpd24/conf.d/mod_cluster.conf`.
 - b. Add the following lines to `mod_cluster.conf`:

```
LoadModule cluster_slotmem_module modules/mod_cluster_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so
```

[Report a bug](#)

2.3.4. Starting JBoss Web Server

To start Red Hat JBoss Web Server, you must start the following:

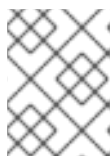
- Tomcat (7 or 8). See [Procedure 2.23, “Starting Tomcat”](#).
- Apache HTTP Server. See [Procedure 2.24, “Starting Apache HTTP Server”](#).

Procedure 2.23. Starting Tomcat

1. In a shell prompt as the root user, start the Tomcat service. Replace `<VERSION>` with the desired Tomcat version (7 or 8):
 - For Red Hat Enterprise Linux 6:

```
# service tomcat<VERSION> start
```
 - For Red Hat Enterprise Linux 7:

```
# systemctl start tomcat<VERSION>.service
```
2. Go to <http://localhost:8080> in your web browser to verify that Tomcat is running.



NOTE

The only supported method of starting Tomcat is using the services shown above.

Procedure 2.24. Starting Apache HTTP Server

- In a shell prompt as the root user, start the Apache HTTP Server service:
 - For Red Hat Enterprise Linux 6:

```
# service httpd24 start
```

- o For Red Hat Enterprise Linux 7:

```
# systemctl start httpd24.service
```

[Report a bug](#)

2.3.5. Stopping JBoss Web Server

To stop Red Hat JBoss Web Server, stop the Tomcat and Apache HTTP Server services.

Procedure 2.25. Stopping Tomcat

1. In a shell prompt as the root user, stop the Tomcat service. Replace `<VERSION>` with the desired Tomcat version (7 or 8):

- o For Red Hat Enterprise Linux 6:

```
# service tomcat<VERSION> stop
```

- o For Red Hat Enterprise Linux 7:

```
# systemctl stop tomcat<VERSION>.service
```

2. Go to <http://localhost:8080> in your web browser to verify that Tomcat is no longer running.

Procedure 2.26. Stopping Apache HTTP Server

- In a shell prompt as the root user, stop the Apache HTTP Server service:

- o For Red Hat Enterprise Linux 6:

```
# service httpd24 stop
```

- o For Red Hat Enterprise Linux 7:

```
# systemctl stop httpd24.service
```

[Report a bug](#)

2.3.6. Configuring JBoss Web Server Services to Start at Boot

You can configure JBoss Web Server (the Tomcat and Apache HTTP Server services) to start at boot.

Use the following command to enable the JBoss Web Server services to start at boot, using the name of your desired Tomcat (either `tomcat7` or `tomcat8`) and Apache HTTP Server (`httpd24`) services.

- For Red Hat Enterprise Linux 6:

```
# chkconfig <SERVICE_NAME> on
```

- For Red Hat Enterprise Linux 7:

```
# systemctl enable <SERVICE_NAME>.service
```

[Report a bug](#)

2.4. SELINUX POLICIES

2.4.1. Default SELinux Policies for an RPM Installation

For each of the RPMs listed in the table below, specific SELinux policies are available. See [Section 2.4.1.1, “Compile SELinux Policies”](#) for an example of compiling and installing the Tomcat policy.

Table 2.1. RPMs and Default SELinux Policies

Name	Port Information	Policy Information
mod_cluster	Two ports (6666 for TCP and 23364 for UDP) are added for httpd_port_t to allow the httpd process to use them.	A post installation script configures the context mapping for /var/cache/mod_cluster to enable the httpd process to write at this location.
tomcat	Four ports are added to http_port_t (TCP ports 8080 , 8005 , 8009 and 8443) to allow the httpd process to use them.	The Tomcat <VERSION> policy is installed, which sets the appropriate SELinux domain for the process when Tomcat executes. It also sets the appropriate contexts to allow tomcat to write to /var/lib/tomcat<VERSION> , /var/log/tomcat<VERSION> , /var/cache/tomcat<VERSION> and /var/run/tomcat<VERSION>.pid .

For more information about using SELinux and other Red Hat Enterprise Linux security information, see the *Red Hat Enterprise Linux Security Guide*

[Report a bug](#)

2.4.1.1. Compile SELinux Policies

You can compile your own Tomcat 7/8 SELinux policies as the root user.

Prerequisites

Before compiling SELinux policies ensure the following prerequisites:

- `selinux-policy-devel` package is installed.
- Tomcat 7 or 8 is installed using the RPM installation method.

Below is an example of a SELinux policy compilation from RPM distributed sources.

Tomcat 7:

```
# cd /etc/tomcat7/selinux/packages/tomcat7
# make -f /usr/share/selinux/devel/Makefile
Compiling targeted tomcat7 module
/usr/bin/checkmodule: loading policy configuration from tmp/tomcat7.tmp
/usr/bin/checkmodule: policy configuration loaded
/usr/bin/checkmodule: writing binary representation (version 10) to

tmp/tomcat7.mod
Creating targeted tomcat7.pp policy package
rm tmp/tomcat7.mod tmp/tomcat7.mod.fc

# semodule -i tomcat7.pp
```

Tomcat 8:

```
# cd /etc/tomcat8/selinux/packages/tomcat8
# make -f /usr/share/selinux/devel/Makefile
Compiling targeted tomcat8 module
/usr/bin/checkmodule: loading policy configuration from tmp/tomcat8.tmp
/usr/bin/checkmodule: policy configuration loaded
/usr/bin/checkmodule: writing binary representation (version 10) to

tmp/tomcat8.mod
Creating targeted tomcat8.pp policy package
rm tmp/tomcat8.mod.fc tmp/tomcat8.mod

# semodule -i tomcat8.pp
```

[Report a bug](#)

2.4.2. SELinux Policies for a ZIP Installation

No SELinux configuration is provided or supported as a default for a Red Hat JBoss Web Server ZIP installation. For a ZIP installation, `httpd` and Tomcat processes run in `httpd_t` or `unconfined_java_t` domains. These domains do not confine the processes, and it is recommended that you undertake the following security precautions:

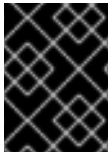
- Always start `httpd` using the `apachectl` script. This ensures that the `apache` user owns the process instead of the `root` user.
- Restrict file access for the `tomcat` and `apache` users to only the files and directories that are necessary to the JBoss Web Server runtime.
- Do not run Tomcat as the `root` user.

[Report a bug](#)

CHAPTER 3. INSTALLING JBOSS WEB SERVER ON MICROSOFT WINDOWS

3.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)

Before installing Red Hat JBoss Web Server on Microsoft Windows, you must first install a supported Java Development Kit (JDK).



IMPORTANT

JBoss Web Server 3 support for Microsoft Windows is limited to the Oracle JDK (1.7 and 1.8) on Windows Server 2012.

Procedure 3.1. Installing Java Development Kit (JDK) for Microsoft Windows

1. Download the Oracle JDK 1.7 or 1.8 for your operating system and architecture. You can download the JDK installation file from the Oracle website:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
2. Double-click the downloaded file to start the Java installation.
3. Proceed as instructed in the installation window.

[Report a bug](#)

3.2. DOWNLOAD AND EXTRACT JBOSS WEB SERVER

To install Red Hat JBoss Web Server, download and extract the installation ZIP files.

Procedure 3.2. Downloading JBoss Web Server

1. Open a browser and log in to the Customer Portal at <http://access.redhat.com>.
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:
 - o Red Hat JBoss Web Server 3.0 Application Server (`jws-application-servers-3.0.0-<platform>-<architecture>.zip`)
 - o Red Hat JBoss Web Server 3.0 Apache HTTP Server (`jws-httpd-3.0.0-<platform>-<architecture>.zip`)

Procedure 3.3. Extract JBoss Web Server

- Unzip the downloaded ZIP files to your installation directory.

**NOTE**

We recommend that you install JBoss Web Server in **C:\Program Files**.

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as *JWS_HOME*.

[Report a bug](#)

3.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION

Some configuration is required before running Red Hat JBoss Web Server. This section includes the following configuration procedures:

- [Procedure 3.4, “Setting Environment Variables”](#)
- [Procedure 3.5, “Running the Post-Installation Scripts”](#)
- [Procedure 3.6, “Installing the Tomcat Service”](#)
- [Procedure 3.7, “Installing the Apache HTTP Server Service”](#)
- [Procedure 3.8, “Enabling log4j Logging for Tomcat”](#)
- [Procedure 3.9, “Removing/Re-Adding SSL Support”](#)
- [Procedure 3.10, “Configuring mod_jk”](#)
- [Procedure 3.11, “Configuring mod_cluster”](#)
- [Procedure 3.12, “Configuring Folder Permissions for JBoss Web Server Services”](#)

Procedure 3.4. Setting Environment Variables

1. Log in to an account with local administrator permissions.
2. Go to **Control Panel** → **System**.
3. Click on the **Advanced** tab.
4. Click the **Environment Variables** button.
5. Click the **New for System Variables**.
6. For **JAVA_HOME**, **TMP** and **TEMP**, enter the appropriate name-value pairs for your system.
7. For the SSL Connector to work, you will also need to add **JWS_HOME\bin** to the **PATH** environment variable of the user that the services will run under. This is **SYSTEM** by default.

Procedure 3.5. Running the Post-Installation Scripts

1. Open a **Command Prompt** with administrator privileges.
2. Change to the **etc** folder of your JBoss Web Server installation:

```
cd /D "JWS_HOME\etc"
```

3. Run the Apache HTTP Server and Tomcat post-installation scripts with the following commands:

```
call postinstall.httpd.bat
```

```
call postinstall.tomcat.bat
```

The scripts create the required symbolic links (Junction Points) for temporary logging and configuration directories.

Procedure 3.6. Installing the Tomcat Service

1. Open a **Command Prompt** with administrator privileges.
2. Change to the **bin** folder for your Tomcat version:

```
cd /D "JWS_HOME\share\tomcat<VERSION>\bin"
```

3. Install the Tomcat service with the following command:

```
call service.bat install
```

Procedure 3.7. Installing the Apache HTTP Server Service



NOTE

By default, Apache HTTP Server is configured to use port 80. If you have Microsoft IIS installed, ensure that you disable or reconfigure it to avoid port conflicts:

- Stop the **World Wide Web . . .** service, and change the Startup Type to **Manual**
- Configure IIS to use different ports.

Alternatively, you can edit **httpd.conf** before installing the Apache HTTP Server service and change **Listen** to a port that does not conflict with the IIS ports.

1. Open a **Command Prompt** with administrator privileges.
2. Change to the **bin** folder of your JBoss Web Server installation:

```
cd /D "JWS_HOME\bin"
```

3. Install the Apache HTTP Server service with the following command:

```
httpd -k install
```

A Firewall security dialog prompt may appear asking for networking access for the Apache HTTP Server. Click **Allow** to access this service from the network.

Procedure 3.8. Enabling log4j Logging for Tomcat

1. Open a **Command Prompt** with administrator privileges.
2. Change to **JWS_HOME\share\extras**
3. Copy the log4j files to the **lib** folder for your Tomcat version:

```
copy log4j-eap6.jar log4j.properties tomcat-juli-adapters.jar
..\tomcat<VERSION>\lib
```

4. Replace **tomcat-juli.jar** file in your Tomcat **bin** directory with the **tomcat-juli.jar** file from **JWS_HOME\share\extras**:

```
copy tomcat-juli.jar ..\tomcat<VERSION>\bin
```

Procedure 3.9. Removing/Re-Adding SSL Support

JBoss Web Server supports SSL by default, but it can be removed. Follow this procedure to remove or re-add SSL support.

1. Open a file browser and go to **JWS_HOME\etc\httpd\conf.d**.
2. Rename the SSL configuration file:
 - o To remove SSL, rename **ssl.conf** to **ssl.conf.disabled**.
 - o To re-add SSL rename **ssl.conf.disabled** to **ssl.conf**.

Procedure 3.10. Configuring mod_jk

If required, you can configure Apache HTTP Server to use mod_jk as its load balancer.

The steps below list the locations of the mod_jk configuration files. For detailed mod_jk configuration instructions, see the *Configure Load Balancing Using Apache HTTP Server and mod_jk* section of the *HTTP Connectors and Load Balancing Guide*.

**NOTE**

Sample configuration files are provided for mod_jk, and are located in **JWS_HOME\etc\httpd\conf.d**. The sample configuration files are: **mod_jk.conf.sample**, **workers.properties.sample**, and **uriworkermap.properties.sample**. To use these samples instead of creating your own configuration files, remove the **.sample** extension, and modify their content as needed.

1. In **JWS_HOME\etc\httpd\conf.d**, create a file named **workers.properties**.

This file should contain the available workers (JBoss instances).

2. In **JWS_HOME\etc\httpd\conf.d**, create a file named **mod_jk.conf**.

This file contains general mod_jk configuration.

3. In `JWS_HOME\etc\httpd\conf.d\`, create a file named `uriworkermap.properties`.

This file contains request mapping rules that map requests to workers.

Procedure 3.11. Configuring `mod_cluster`

The `mod_cluster` load balancer is optional. Follow this procedure to configure Apache HTTP Server to use `mod_cluster` as its load balancer.

For more information about customizing `mod_cluster`, see the *mod_cluster Connector* section of the *HTTP Connectors and Load Balancing Guide*

1. In `JWS_HOME\etc\httpd\conf.modules.d\00-proxy.conf`, ensure that there is a `#` at the beginning of the following line to disable `mod_proxy_balancer`.so:

```
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

This module is incompatible with `mod_cluster`.

2. Configure the server to load the `mod_cluster` modules:
 - a. Create the file `JWS_HOME\etc\httpd\conf.d\mod_cluster.conf`.
 - b. Add the following lines to `mod_cluster.conf`:

```
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule cluster_slotmem_module modules/mod_cluster_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule advertise_module modules/mod_advertise.so
```

Procedure 3.12. Configuring Folder Permissions for JBoss Web Server Services

Follow this procedure to ensure that the account used to run the services has full control over the `JWS_HOME` folder and all of its subfolders:

1. Right-click the `JWS_HOME` folder and click **Properties**.
2. Select the **Security** tab.
3. Click the **Edit** button.
4. Click the **Add** button.
5. In the text box, enter **LOCAL SERVICE**.
6. Select the **Full Control** check box for the **LOCAL SERVICE** account.
7. Click **OK**.
8. Click the **Advanced** button.
9. Inside the **Advanced Security Settings** dialog, select **LOCAL SERVICE** and click **Edit**.
10. Select the check box next to the **Replace all existing inheritable permissions on all descendants with inheritable permissions from this object** option.

11. Click **OK** through all the open folder property windows to apply the settings.

[Report a bug](#)

3.4. STARTING JBOSS WEB SERVER

To start Red Hat JBoss Web Server, you must start the following:

- Tomcat
- Apache HTTP Server

You can start the services from the Command Prompt, or with the Computer Management tool.

Procedure 3.13. Starting JBoss Web Server from the Command Prompt

Follow this procedure to start the Apache HTTP Server and Tomcat services from the Command Prompt.

1. Open a **Command Prompt** with administrator privileges.
2. Start the Tomcat service:

```
net start tomcat<VERSION>
```

3. Start the Apache HTTP Server (httpd) service:

```
net start Apache2.4
```

Procedure 3.14. Starting JBoss Web Server from the Computer Management Tool

Follow this procedure to start the Apache HTTP Server and Tomcat services from the Computer Management tool.

1. Go to **Start** → **Administrative Tools** → **Services**
2. In the Services list, right-click the name of each service (`httpd` and `tomcat`) and click **Start**.

NOTE

Some third-party applications add libraries to the system directory in Windows. These take precedence over Tomcat libraries when looked-up. This means that if those third-party libraries have the same name as the those used by Tomcat native libraries, they are loaded instead of the libraries distributed with JBoss Web Server.

In this situation, Tomcat may not start, and does not log any error messages in the Windows Event Log, or Tomcat log files. Errors can only be seen by using `catalina.bat run`.

If this behavior occurs, inspect the contents of the `C:\windows\System32\` directory and other PATH directories, and ensure that there are no DLLs conflicting with those delivered with JBoss Web Server. In particular, look for `libeay32.dll`, `ssleay32.dll`, and `libssl32.dll`.

[Report a bug](#)

3.5. STOPPING JBOSS WEB SERVER

To stop Red Hat JBoss Web Server, you must stop the following:

- Tomcat
- Apache HTTP Server

You can stop the services from the Command Prompt, or with the Computer Management tool.

Procedure 3.15. Stopping JBoss Web Server from the Command Prompt

Follow this procedure to stop the Apache HTTP Server and Tomcat services from the Command Prompt.

1. Open a **Command Prompt** with administrator privileges.
2. Stop the Tomcat service:

```
net stop tomcat<VERSION>
```

3. Stop the Apache HTTP Server (httpd) service:

```
net stop Apache2.4
```

Procedure 3.16. Stopping JBoss Web Server from the Computer Management Tool

Follow this procedure to stop the Apache HTTP Server and Tomcat services from the Computer Management tool.

1. Go to **Start** → **Administrative Tools** → **Services**
2. In the Services list, right-click the name of each service (**httpd** and **tomcat**) and click **Stop**.

[Report a bug](#)

CHAPTER 4. INSTALLING JBOSS WEB SERVER ON SOLARIS

4.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)

Before installing Red Hat JBoss Web Server on Solaris, you must first install a supported Java Development Kit (JDK).



IMPORTANT

JBoss Web Server 3 support for Solaris is limited to the Oracle JDK (1.7 and 1.8) on Solaris 10 and Solaris 11.

Procedure 4.1. Installing Java

1. Download the Oracle JDK 1.7 or 1.8 for your operating system and architecture. You can download the JDK installation file from the Oracle website: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
2. Run the Java installation file.
3. Open `/usr` at a shell prompt, and run the following command to display the current Java symbolic link:

```
ls -lad java
```

4. Remove the link:

```
rm java
```

5. Create a Java symbolic link to the newly installed JDK:

```
ln -sf /usr/jdk/<JDK>
```

[Report a bug](#)

4.2. DOWNLOAD AND EXTRACT JBOSS WEB SERVER

To install Red Hat JBoss Web Server, download and extract the installation ZIP files.

Procedure 4.2. Downloading JBoss Web Server

1. Open a browser and log in to the Customer Portal at <http://access.redhat.com>.
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select the correct JBoss Web Server version from the **Version** drop-down menu.
5. Click **Download** for each of the following files, ensuring that you select the correct platform and architecture for your system:
 - o Red Hat JBoss Web Server 3.0 Application Server (`jws-application-servers-`

3.0.0-<platform>-<architecture>.zip)

- o Red Hat JBoss Web Server 3.0 Apache HTTP Server (jws-httpd-3.0.0-<platform>-<architecture>.zip)

Procedure 4.3. Extract JBoss Web Server

- Unzip the downloaded ZIP files to your installation directory.



NOTE

We recommend that you install JBoss Web Server in the `/opt/` directory.

The directory created by extracting the ZIP archives is the top-level directory for JBoss Web Server. This is referred to as `JWS_HOME`.

[Report a bug](#)

4.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION

Some configuration is required before running Red Hat JBoss Web Server. This section includes the following configuration procedures:

- [Procedure 4.4, “Running the Post-Installation Scripts”](#)
- [Procedure 4.5, “Setting the JAVA_HOME Environment Variable”](#)
- [Procedure 4.6, “Removing/Re-Adding SSL Support”](#)
- [Procedure 4.7, “Enabling log4j Logging for Tomcat”](#)
- [Procedure 4.8, “Configuring mod_jk”](#)
- [Procedure 4.9, “Configuring Apache HTTP Server to Use mod_cluster”](#)

Procedure 4.4. Running the Post-Installation Scripts

1. Open a shell prompt, and change directory to `JWS_HOME/etc`.
2. Issue the following commands to run the post-installation scripts:

```
sh .postinstall.httpd
sh .postinstall.tomcat
```

The post-installation scripts create the Apache and Tomcat users and groups.

Procedure 4.5. Setting the JAVA_HOME Environment Variable

You must set the `JAVA_HOME` environment variable for Tomcat before running JBoss Web Server.

1. Open the Tomcat configuration file:
 - o For Tomcat 7: `JWS_HOME/etc/sysconfig/tomcat7`

- o For Tomcat 8: `JWS_HOME/etc/sysconfig/tomcat8`
2. Remove the hash (#) sign at the beginning of the following line:

```
# JAVA_HOME="/usr/java"
```

Procedure 4.6. Removing/Re-Adding SSL Support

JBoss Web Server supports SSL by default, but it can be removed. Follow this procedure to remove or re-add SSL support.

- Go to `JWS_HOME/etc/httpd/conf.d` and rename the SSL configuration file:
 - o To remove SSL, rename `ssl.conf` to `ssl.conf.disabled`.
 - o To re-add the SSL, rename `ssl.conf.disabled` to `ssl.conf`.

Procedure 4.7. Enabling log4j Logging for Tomcat

If required, follow this procedure to add log4j logging to Tomcat.

1. Open a shell prompt and change directory to `JWS_HOME/share/extras/`.
2. Copy the `log4j-eap6.jar`, `log4j.properties`, and `tomcat-juli-adapters.jar` files to the `lib` directory of the Tomcat directory.

For example:

```
extras]# cp log4j.properties ../tomcat<VERSION>/lib/
```

```
extras]# cp log4j-eap6.jar ../tomcat<VERSION>/lib/
```

```
extras]# cp tomcat-juli-adapters.jar ../tomcat<VERSION>/lib/
```

Replace `<VERSION>` with the respective Tomcat version number (7 or 8).

3. Replace `tomcat-juli.jar` file in your Tomcat `bin` directory with the `tomcat-juli.jar` file from `JWS_HOME/share/extras/`:

```
extras]# cp tomcat-juli.jar ../tomcat<VERSION>/bin/
```

Procedure 4.8. Configuring mod_jk

If required, you can configure Apache HTTP Server to use `mod_jk` as its load balancer.

The steps below list the locations of the `mod_jk` configuration files. For detailed `mod_jk` configuration instructions, see the *Configure Load Balancing Using Apache HTTP Server and mod_jk* section of the *HTTP Connectors and Load Balancing Guide*.

**NOTE**

Sample configuration files are provided for `mod_jk`, and are located in `JWS_HOME/etc/httpd/conf.d/`. The sample configuration files are: `mod_jk.conf.sample`, `workers.properties.sample`, and `uriworkermap.properties.sample`. To use these samples instead of creating your own configuration files, remove the `.sample` extension, and modify their content as needed.

1. In `JWS_HOME/etc/httpd/conf.d/`, create a file named `workers.properties`.

This file should contain the available workers (JBoss instances).

2. In `JWS_HOME/etc/httpd/conf.d/`, create a file named `mod_jk.conf`.

This file contains general `mod_jk` configuration.

3. In `JWS_HOME/etc/httpd/conf.d/`, create a file named `uriworkermap.properties`.

This file contains request mapping rules that map requests to workers.

Procedure 4.9. Configuring Apache HTTP Server to Use `mod_cluster`

- `mod_cluster` can be configured in `JWS_HOME/httpd/conf.d/mod_cluster.conf`.

For more information about customizing `mod_cluster`, see the *`mod_cluster` Connector* section of the *HTTP Connectors and Load Balancing Guide*

[Report a bug](#)

4.4. STARTING JBOSS WEB SERVER

To start Red Hat JBoss Web Server, you must start the following:

- Tomcat (7 or 8).
- Apache HTTP Server.

Procedure 4.10. Starting Tomcat

- Run the following command with your respective Tomcat version (7 or 8):

```
$ sh JWS_HOME/share/apache-tomcat-<VERSION>/bin/daemon.sh
```

**IMPORTANT**

Although there are multiple methods of starting Tomcat, it is recommended that you use the `daemon.sh` script. To start Tomcat as a service using `Jsvc`, see [Chapter 5, Using `Jsvc` to Start Tomcat](#).

Procedure 4.11. Starting Apache HTTP Server

- To start Apache HTTP Server (httpd), in a terminal as the root, change to *JWS_HOME/sbin/* and run the following command:

```
# ./apachectl start
```

[Report a bug](#)

4.5. STOPPING JBOSS WEB SERVER

To stop Red Hat JBoss Web Server, you must stop the following:

- Tomcat (7 or 8).
- Apache HTTP Server.

Procedure 4.12. Stopping Tomcat

- To stop Tomcat, in a shell prompt as the root user change to *JWS_HOME/sbin/* and run the following command with your respective Tomcat version (7 or 8):

```
tomcat<VERSION> stop
```

Procedure 4.13. Stopping Apache HTTP Server

- To stop Apache HTTP Server, in a shell prompt as the root user change to *JWS_HOME/sbin/* and run the following command:

```
apachectl stop
```

[Report a bug](#)

CHAPTER 5. USING JSVC TO START TOMCAT

Jsvc is a set of libraries and applications that facilitates running Java applications on Linux, UNIX, and similar operating systems. Using Jsvc with Red Hat JBoss Web Server 3 allows Tomcat to switch identities. Using Jsvc, Tomcat can perform root-level operations and then revert to a non-privileged user. Jsvc is primarily used for running Tomcat as a service.

Jsvc files are available at the following locations:

For Red Hat Enterprise Linux:

- `JWS_HOME/extras/jsvc`
- `JWS_HOME/tomcat<VERSION>/bin/jsvc`



NOTE

`JWS_HOME/tomcat<VERSION>/bin/jsvc` is a symlink to `JWS_HOME/extras/jsvc`.

For Solaris:

- `JWS_HOME/sbin/jsvc`
- `JWS_HOME/share/apache-tomcat-<VERSION>/bin/jsvc`



NOTE

`JWS_HOME/share/apache-tomcat-<VERSION>/bin/jsvc` is a symlink to `JWS_HOME/sbin/jsvc`.

[Report a bug](#)

5.1. STARTING TOMCAT USING JSVC

Procedure 5.1. Start Tomcat Using Jsvc on Red Hat Enterprise Linux

- Run the following command to start Tomcat using Jsvc on Red Hat Enterprise Linux:

```
JWS_HOME/tomcat<VERSION>/bin/daemon.sh start
```

Procedure 5.2. Start Tomcat Using Jsvc on Solaris

- Run the following command to start Tomcat using Jsvc on Solaris:

```
JWS_HOME/share/tomcat<VERSION>/bin/daemon.sh start
```

[Report a bug](#)

5.2. STOPPING TOMCAT USING JSVC

Procedure 5.3. Stop Tomcat Using Jsvc on Red Hat Enterprise Linux

Procedure 5.3. Stop Tomcat Using Jsvc on Red Hat Enterprise Linux

- Run the following command to stop Tomcat that was started using Jsvc on Red Hat Enterprise Linux:

```
JWS_HOME/tomcat<VERSION>/bin/daemon.sh stop
```

Procedure 5.4. Stop Tomcat Using Jsvc on Solaris

- Run the following command to stop Tomcat that was started using Jsvc on Solaris:

```
JWS_HOME/share/tomcat<VERSION>/bin/daemon.sh stop
```

[Report a bug](#)

5.3. JSVC PARAMETERS

The following parameters can be configured when running the `daemon.sh` script:

Table 5.1. daemon.sh Startup Parameters

Parameter Name	Environment Variable	Default Value	Description
<code>--java-home</code>	<code>JAVA_HOME</code>	Based on the value of the <code>PATH</code> variable.	The Java home directory location.
<code>--catalina-home</code>	<code>CATALINA_HOME</code>	Determined by the location of the script.	The Tomcat installation directory location.
<code>--catalina-base</code>	<code>CATALINA_BASE</code>	Based on the value of the <code>PATH</code> variable.	The directory that contains the specific configuration and set up information if multiple servers are using the same installation.
<code>--catalina-pid</code>	-	<code>\$CATALINA_BASE/logs/catalina-daemon.pid</code>	The file where the process ID (PID) for the running instance of Tomcat is stored.
<code>--tomcat-user</code>	-	<code>tomcat</code>	The user Tomcat uses.
<code>--service-start-wait-time</code>	-		This is a wrapper to the <code>--wait</code> parameter. The <code>--wait</code> parameter accepts values in seconds.

[Report a bug](#)

CHAPTER 6. HIBERNATE ON JBOSS WEB SERVER

Hibernate is an object-relational mapping framework. It is packaged independently from Red Hat JBoss Web Server. This packaged version is used on all supported platforms.

Hibernate is used in the same way it is used for a regular Tomcat installation: the Hibernate JAR files are added into a deployment WAR file. Tomcat provides a default connection pooling mechanism, which is defined in `context.xml`. However, `persistence.xml` and `web.xml` are also required. The example below shows a configuration with the Tomcat connection pooling mechanism.

- `/META-INF/context.xml` defines the connection pools Tomcat should create.

Example 6.1. context.xml

```
<Context>
  <Resource
    name="jdbc/DsWebAppDB"
    auth="Container"
    type="javax.sql.DataSource"
    username="sa"
    password=""
    driverClassName="org.h2.Driver"
    url="jdbc:h2:mem:target/test/db/h2/hibernate"
    maxActive="8"
    maxIdle="4"/>
</Context>
```

- `/WEB-INF/classes/META-INF/persistence.xml` is a JPA configuration file. It defines how the application configures Hibernate to consume connections from the Tomcat pool. If you are using the Hibernate API directly, use a similar configuration to that shown in `hibernate.cfg.xml`.

Example 6.2. persistence.xml

```
<persistence version="1.0"
  xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">

  <persistence-unit name="dswebapp">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>
    <properties>
      <property name="hibernate.dialect"
value="org.hibernate.dialect.H2Dialect" />
      <property name="hibernate.connection.datasource"
value="java:comp/env/jdbc/DsWebAppDB"/>
    </properties>
  </persistence-unit>
</persistence>
```

- `/WEB-INF/web.xml` is a regular web application deployment file, which instructs Tomcat which datasource to consume. In the example below, the datasource is `jdbc/DsWebAppDB`.

Example 6.3. web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <resource-env-ref>
    <resource-env-ref-name>jdbc/DsWebAppDB</resource-env-ref-name>
    <resource-env-ref-type>javax.sql.DataSource</resource-env-ref-
type>
  </resource-env-ref>
</web-app>
```

For details, see the Hibernate documentation for JBoss Web Server.

[Report a bug](#)

CHAPTER 7. MONITORING JBOSS WEB SERVER WITH JBOSS OPERATIONS NETWORK (ON)

Red Hat JBoss Web Server can be monitored and managed by Red Hat JBoss Operations Network (JBoss ON).

To monitor JBoss Web Server using JBoss ON, you must install the **Web Server Plugin Pack** on your JBoss ON server, and then configure Apache HTTP Server and Tomcat in JBoss Web Server to be monitored.

[Report a bug](#)

7.1. DOWNLOAD THE WEB SERVER PLUGIN PACK FOR JBOSS ON

To monitor Red Hat JBoss Web Server using JBoss ON, you must first download and install the **Web Server Plugin Pack** on your JBoss ON server.

Procedure 7.1. Download the Web Server Plugin Pack for JBoss ON

1. Open a web browser, and log in to the Red Hat Customer Portal: <http://access.redhat.com>
2. Click **Downloads**.
3. Click **Red Hat JBoss Web Server** in the **Product Downloads** list.
4. Select **JBoss ON for Web Server** in the **Product** drop-down menu.
5. Click **Download** for each of the following files:
 - o **Web Server Plugin Pack for Red Hat JBoss Operations Network**

Consult the JBoss ON documentation for instructions on installing the plugin on your JBoss ON server.

[Report a bug](#)

7.2. CONFIGURING APACHE HTTP SERVER FOR JBOSS ON MONITORING

To allow JBoss ON to monitor Apache HTTP Server in Red Hat JBoss Web Server, you must enable the `mod_bmx` modules.

Procedure 7.2. Configuring Apache HTTP Server for JBoss ON Monitoring

1. As the `apache` user, enable the `mod_bmx` modules. Create the file `JWS_HOME/httpd/conf.d/mod_bmx.conf` with the following contents:

```
LoadModule bmx_module modules/mod_bmx.so
LoadModule bmx_status_module modules/mod_bmx_status.so
LoadModule bmx_vhost_module modules/mod_bmx_vhost.so
```

2. Add a handler for `mod_bmx`. In `JWS_HOME/httpd/conf/httpd.conf`, append your handler definition. For example:

■


```
<Location /bmx>
SetHandler bmx-handler
</Location>
```

You may want to restrict access to this URL to a specific virtual host or use password authentication.

- Restart the Apache HTTP Server. You can access `http://SERVER_ADDRESS:PORT/bmx` in a web browser to verify that `mod_bmx` is working correctly.
- The Apache HTTP Server in your JBoss Web Server installation will now be discoverable for JBoss ON. In JBoss ON, you can import it from the Discovery Queue.

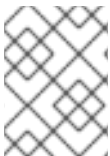
Consult the JBoss ON documentation for further details on monitoring and managing Apache HTTP Server.

[Report a bug](#)

7.3. CONFIGURING TOMCAT FOR JBOSS ON MONITORING

To allow JBoss ON to monitor Tomcat in Red Hat JBoss Web Server, you must configure Tomcat to allow JBoss ON discovery, as well as providing JBoss ON the required access.

Procedure 7.3. Configuring User Permissions on a Linux Operating System



NOTE

For Microsoft Windows, skip these steps and proceed to [Procedure 7.4, “Configuring Tomcat for JBoss ON Monitoring”](#).

The JBoss ON agent requires read and write permission to the `httpd` and `Tomcat` directories.

- As a user with root privileges, run the following command to add the user which runs the JBoss ON Agent to the `tomcat` and `apache` user groups:

```
# usermod -aG tomcat,apache <JBOSSON_AGENT_USER>
```

Procedure 7.4. Configuring Tomcat for JBoss ON Monitoring

JBoss Web Server instances are auto-discovered on Linux and Unix platforms. However, you need to configure the instance's JMX to allow for proper handling of authentication, and accurate Tomcat monitoring.

To configure JMX to handle authentication:

- Open the `startup` file of the respective JBoss Web Server instance for editing:
 - On Red Hat Enterprise Linux installed from a `ZIP` file, open `JWS_HOME/tomcat<VERSION>/bin/startup.sh`
 - On Red Hat Enterprise Linux installed from `RPM`, open `/usr/sbin/tomcat<VERSION>`
 - On Microsoft Windows open `JWS_HOME\share\tomcat<VERSION>\bin\startup.bat`.

- o On Solaris using `daemon.sh` to start Tomcat, open `JWS_HOME/tomcat<VERSION>/bin/setenv.sh`
2. Define an available port for JMX monitoring. Ensure the port is not blocked by any firewall.

- o On Red Hat Enterprise Linux and Solaris:

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.port=PORT_NUMBER -
Djava.rmi.server.hostname=IP_ADDRESS"
```

- o On Microsoft Windows:

```
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.port=PORT_NUMBER -
Djava.rmi.server.hostname=IP_ADDRESS"
```

3. In production environments, add the following lines to the `JAVA_OPTS` variable in the startup file to secure JMX with SSL and restrict the access with a firewall:

- o On Red Hat Enterprise Linux and Solaris

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.access.file=JWS_HOME/jmxremote.access"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.password.file=JWS_HOME/jmxremote.password"
```

- o On Microsoft Windows:

```
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.access.file=JWS_HOME\jmxremote.access"
set "JAVA_OPTS=%JAVA_OPTS% -
Dcom.sun.management.jmxremote.password.file=JWS_HOME\jmxremote.password"
```



NOTE

If you want to disable authentication and SSL for development purposes, add the following lines to the `JAVA_OPTS` variable in the startup file:

```
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.authenticate=false"
```

4. Once the Tomcat server resource is discovered and imported into the JBoss ON inventory, it may be necessary to update the new resource's connection settings.

- a. In the JBoss ON interface, click **Connection Settings** for the newly imported Tomcat server resource.

- b. Verify the value of the **Manager URL** property to the RMI URL, to ensure it uses the correct JMX host name and port number as defined in the Tomcat server startup file. An example for this value is shown below:

```
service:jmx:rmi:///jndi/rmi://$IP_ADDRESS:$PORT/jmxrmi
```

[Report a bug](#)

7.3.1. Configuring JBoss ON Monitoring for Tomcat Installed from RPMs

Procedure 7.5. Configuring JBoss ON Monitoring for Tomcat Installed from RPMs

1. In a shell prompt become the root user.
2. Set up JMX JAVA_OPTS properties in the `/usr/sbin/tomcat<VERSION>/startup` file in the `start` and `start-security` sections.

```
if [ "$1" = "start" ]; then
  JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.port=8100 -
  Dcom.sun.management.jmxremote.ssl=false -
  Dcom.sun.management.jmxremote.authenticate=true -
  Dcom.sun.management.jmxr
  emote.access.file="/etc/tomcat<VERSION>/jmxremote.access" -
  Dcom.sun.management.jmxremote.password.file="/etc/tomcat<VERSION>/jm
  xremote.password"

  ${JAVACMD} $JAVA_OPTS $LOGGING_CONFIG $CATALINA_OPTS \
  -classpath "$CLASSPATH" \
  -Dcatalina.base="$CATALINA_BASE" \
  -Dcatalina.home="$CATALINA_HOME" \
  -Djava.endorsed.dirs="$JAVA_ENDORSED_DIRS" \
  -Djava.io.tmpdir="$CATALINA_TMPDIR" \
  org.apache.catalina.startup.Bootstrap start \
  >> ${CATALINA_BASE}/logs/catalina.out 2>&1 &
  if [ ! -z "$CATALINA_PID" ]; then
    echo $! > $CATALINA_PID
  fi
elif [ "$1" = "start-security" ]; then
  JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.port=8100 -
  Dcom.sun.management.jmxremote.ssl=false -
  Dcom.sun.management.jmxremote.authenticate=true -
  Dcom.sun.management.jmxr
  emote.access.file="/etc/tomcat<VERSION>/jmxremote.access" -
  Dcom.sun.management.jmxremote.password.file="/etc/tomcat<VERSION>/jm
  xremote.password"

  ${JAVACMD} $JAVA_OPTS $LOGGING_CONFIG $CATALINA_OPTS \
  -classpath "$CLASSPATH" \
  -Dcatalina.base="$CATALINA_BASE" \
```

3. In `/etc/tomcat<VERSION>/tomcat<VERSION>.conf`, configure Tomcat to run as the root user.

```
TOMCAT_USER="root"
```

-
4. Run the following command to start Tomcat.

```
service tomcat<VERSION> start
```

5. Start the JBoss ON agent.
6. In the JBoss ON Web UI, import the JBoss ON agent and Tomcat.
7. In the JBoss ON Web UI, setup the Tomcat connection configuration (principal and credentials).
8. In the JBoss ON Web UI, set the Tomcat Control method configuration to **RPM System V init script**.



NOTE

Start and Shutdown script may not be set because the Tomcat plugin always runs the `service tomcat<VERSION> start/stop` command for the **RPM System V init script** configuration setting.

[Report a bug](#)

7.3.2. Configuring JBoss ON Monitoring for Tomcat Installed as a Windows Service

Procedure 7.6. Configuring JBoss ON for JBoss Web Server with Tomcat Installed as a Windows Service

1. Create the `jmxremote.access` file with *controlRole readwrite* in the `C:\jmx` directory.
2. Create the `jmxremote.password` file with *controlRole pwd* in the `C:\jmx` directory.



NOTE

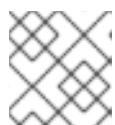
Set the owner of `jmxremote.access` and `jmxremote.password` to `SYSTEM`, and restrict the access of `jmxremote.password` only to `SYSTEM`. The `SYSTEM` user must only have read access.

3. Enable JMX for the Tomcat Windows service.

```
JWS_HOME\sbin\tomcat<VERSION>.exe //US//Tomcat<VERSION>
++JvmOptions="-Dcom.sun.management.jmxremote.port=8100;-
Dcom.sun.management.jmxremote.access.file="C:\jmx\jmxremote.access";
-
Dcom.sun.management.jmxremote.password.file="C:\jmx\jmxremote.passwo
rd";-Dcom.sun.management.jmxremote.ssl=false;-
Dcom.sun.management.jmxremote.authenticate=true"
```

4. Start the Tomcat Windows service, and verify that it is running.
5. Install and configure the JBoss ON agent. Type **discovery** in the agent prompt to discover the Tomcat Windows service.

6. In the JBoss ON Web UI, click **Inventory** and then, click **Discovery Queue** and select **import Tomcat and RHQ agent**.
7. In the JBoss ON Web UI, go to **Plat forms** and search for the agent name. Click on your agent.
8. On the **Agent** page, Tomcat Servers are listed. Select your Tomcat server by clicking it.
9. In the JBoss ON Web UI, click on the **Inventory** tab and then configure the Tomcat Server in **Connection Settings**.
10. Enter the Principal and Credentials information. use the *controlRole* and password set in the *jmxremote* files.
11. Set the control method to **RPM System V init script**.

**NOTE**

You can not set **Start** and **Shutdown Script** fields.

12. Click **Save**.
13. Update the connection settings of the Tomcat Server JVM and set Principal and Credentials.

[Report a bug](#)

APPENDIX A. SUPPORTED APACHE HTTP SERVER MODULES

Supported Apache HTTP Server modules are included with Red Hat JBoss Web Server. You can find the module binaries in the following location for your respective operating system:

- Red Hat Enterprise Linux: ***JWS_HOME*/httpd/modules**
- Microsoft Windows: ***JWS_HOME*\lib64\httpd\modules**
- Solaris: ***JWS_HOME*/etc/httpd/modules**

[Report a bug](#)

APPENDIX B. REVISION HISTORY

Revision 3.0.3-22	Monday 4 July 2016	Lucas Costi
Red Hat JBoss Web Server 3.0.3 Asynchronous Documentation Update: Includes details of Red Hat Network channels for Red Hat Satellite 5. mod_ssl package name fix for JWS-477 .		
Revision 3.0.1-18	Tuesday 22 September 2015	Lucas Costi
Red Hat JBoss Web Server 3.0.1 Asynchronous Documentation Update.		
Revision 3.0.1-17	Friday 4 September 2015	Lucas Costi
Red Hat JBoss Web Server 3.0.1 GA.		
Revision 3.0.0-14	Tue 28 July 2015	Lucas Costi
Added 'Note' for JWS-187 .		
Revision 3.0.0-12	Tue 5 May 2015	Lucas Costi
Red Hat JBoss Web Server 3.0 GA.		