Red Hat JBoss Web Server 5.5

Installation Guide

Install and Configure Red Hat JBoss Web Server 5.5

Last Updated: 2021-11-08

Red Hat JBoss Web Server 5.5 Installation Guide

Install and Configure Red Hat JBoss Web Server 5.5

Legal Notice

Copyright © 2021 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Abstract

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

Table of Contents

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. INTRODUCTION

This Installation Guide includes procedures for the installation, minor upgrade, and basic configuration of the Tomcat servers from JBoss Web Server on supported operating systems. The installation and configuration instructions for the Apache HTTP Server are available in the JBoss Core Services Documentation.

1.1. ABOUT RED HAT JBOSS WEB SERVER

Red Hat JBoss Web Server is a fully integrated and certified set of components for hosting Java web applications. It consists of the following components:

- an application server (Apache Tomcat Servlet container),
- Tomcat Native Library

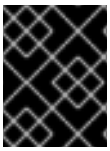
1.1.1. Full List of Components

Red Hat JBoss Web Server contains the following components:

Component	Version
Apache CXF	3.3.5
Apache Tomcat 9	9.0.43
ECJ	4.12.0
Hibernate	5.3.20.Final
JBoss logging	3.4.1.Final
libapr	1.6.3
mod_cluster	1.4.3.Final
OpenSSL	1.1.1g
Tomcat-Native	1.2.26
Tomcat-Vault	1.1.8.Final

IMPORTANT

Apache Tomcat is also provided by the RHEL platform subscription as part of RHEL 7, but **NOT** RHEL 8. In the future releases, Tomcat will be available as a part of the Middleware Runtimes subscription. There are some differences in the Tomcat provided by RHEL and Tomcat provided by the Runtimes JWS subscription. RHEL 7 has Tomcat 7. JWS version 3.1 provides Tomcat 7 and 8 and JWS version 5.x provides Tomcat 9. Additionally, both projects provide RPM packages, but only JWS provides .ZIP archives.



IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

The description for some key components are:

- **Apache tomcat**: a servlet container in accordance with the Java Servlet Specification. JBoss Web Server contains Apache Tomcat 9.
- **Apache tomcat native library**: a Tomcat library that improves Tomcat scalability, performance, and integration with native server technologies.
- **tomcat-vault**: an extension for the JBoss Web Server used for securely storing passwords and other sensitive information used by a JBoss Web Server.
- **mod_cluster** library: a library that allows communication between Apache Tomcat and the **mod_proxy_cluster** module of Apache HTTP Server. This enables you to use the Apache HTTP Server as a load balancer for JBoss Web Server. For more information about the configuration of **mod_cluster**, or for information about the installation and configuration of the alternative load balancers **mod_jk** and **mod_proxy**, see the HTTP Connectors and Load Balancing Guide .
- **Apache portable runtime (APR)**: A runtime that provides superior scalability, performance, and improved integration with native server technologies. APR is a highly portable library that is at the heart of Apache HTTP Server 2.x. It enables access to:
  - Advanced IO functionality - For example, sendfile, epoll and OpenSSL
  - Operating system level functionality - For example, random number generation and system status
  - Native process handling - For example, shared memory, NT pipes and Unix sockets
- **OpenSSL**: A software library that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols and includes a basic cryptographic library.

NOTE

- 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).

1.2. SUPPORTED OPERATING SYSTEMS AND CONFIGURATIONS

Red Hat JBoss Web Server supports the following operating systems and configurations .

1.3. METHODS TO INSTALL RED HAT JBOSS WEB SERVER

You can install JBoss Web Server on supported Red Hat Enterprise Linux and Microsoft Windows systems using archive installation files available for each platform. You can also install JBoss Web Server on supported Red Hat Enterprise Linux systems using RPM packages.

The following components are included in the archive installation files. These components are the core parts of a JBoss Web Server installation.

- **jws-5.5.0-application-server.zip**
  - Tomcat 9
  - mod_cluster
  - tomcat-vault
- **jws-5.5.0-application-server-*<platform>*-*<architecture>*.zip**
  - Platform-specific utilities

1.4. COMPONENT DOCUMENTATION BUNDLE

JBoss Web Server includes an additional documentation bundle that includes the original vendor documentation for each component. This documentation bundle, **jws-docs-5.5.0.zip**, is available at the Red Hat Customer Portal, and contains additional documentation for the following components:

- tomcat
- tomcat-native
- tomcat-vault

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

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

- Archive files
- RPM packages

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

Prerequisites for Red Hat Enterprise Linux-7 and Red Hat Enterprise Linux-8 are different, see Red Hat Enterprise Linux Package Prerequisites.

2.1. PREREQUISITES

2.1.1. Installing a Java Development Kit (JDK) using the YUM package manager

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

For a completed list of supported JDKs see Supported operating systems and configurations .

Procedure

1. Subscribe your Red Hat Enterprise Linux system to the appropriate channel:
   - **OpenJDK:**
     - rhel-7-server-rpms
     - rhel-8-server-rpms
   - **IBM:**
     - rhel-7-server-supplementary-rpms
     - rhel-8-server-supplementary-rpms

IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

1. As the root user, execute the command to install a 1.8 JDK:

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

   Replace ***<VENDOR>*** with **ibm** or **openjdk**

2. 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.

2.1.2. Installing a JDK from a compressed archive (such as .zip or .tar)

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

A full list of supported JDKs is given in section 1.2 of this document.

If the JDK was downloaded from the vendor's website (Oracle or OpenJDK), use the installation instructions provided by the vendor and set the **JAVA_HOME** environment variable.

If the JDK has was installed from a compressed archive, set the **JAVA_HOME** environment variable for Tomcat before running JBoss Web Server.

In the **bin** directory of Tomcat (***JWS_HOME*/tomcat/bin**), create a file named **setenv.sh**, and insert the **JAVA_HOME** path definition.

For example:

```
$ cat JWS_HOME/tomcat/bin/setenv.sh

export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk.x86_64
```

2.1.3. Red Hat Enterprise Linux Package Prerequisites

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

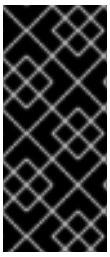
- A supported JDK is installed.
- Additionally, **RHEL 8** users needing OpenSSL or APR need to install them from the operating system. To install OpenSSL and APR, run the following commands:

```
# yum install openssl
```

```
# yum install apr
```

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

```
# yum remove tomcatjss
```



IMPORTANT

- In **RHEL 7**, JWS uses OpenSSL and APR from Red Hat JBoss Core Services however in **RHEL 8** OpenSSL and APR are used from the operating system.
- **RHEL 8** zip package does not contain OpenSSL and APR which should be installed from the operating system.

2.2. INSTALLING AND MANAGING JBOSS WEB SERVER (ZIP)

You can install JBoss Web Server from an archive file. Installation from an archive results in different methods of managing the product compared to installation from an RPM package. For example, you can use a system daemon at boot time and manage JBoss Web Server from a command line. Start by downloading and extracting the archive file.

2.2.1. Downloading and Extracting JBoss Web Server

This method of installation involves accessing the Red Hat Customer Portal and locating the correct version of JBoss Web Server.

Prerequisites

- Ensure that all of the prerequisites are met before installing JBoss Web Server.

Procedure

To install JBoss Web Server, download and extract the installation archive files.

1. Open a browser and log in to the Red Hat Customer Portal .
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:
   - The Red Hat JBoss Web Server 5.5 Application Server (**jws-5.5.0-application-server.zip**).
   - The Red Hat JBoss Web Server 5.5 Native Components for RHEL (**jws-5.5.0-application-server-*<platform>*-*<architecture>*.zip**).
6. Unzip the downloaded archive files to your installation directory.
   For example:

```
# unzip jws-5.5.0-application-server.zip -d /opt/
# unzip -o jws-5.5.0-application-server-<platform>-<architecture>.zip -d /opt/
```

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

2.2.2. Managing JBoss Web Server on Red Hat Enterprise Linux

There are two supported methods for running and managing Red Hat JBoss Web Server on Red Hat Enterprise Linux:

- using a system daemon
- on a command line

The recommended method for managing the JBoss Web Server is using a system daemon.

2.2.2.1. Managing JBoss Web Server using a system daemon for .zip installations on Red Hat Enterprise Linux

Using the JBoss Web Server with a system daemon provides a method of starting the JBoss Web Server services at system boot. The system daemon also provides start, stop and status check functions.

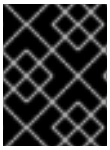
The default system daemon for Red Hat Enterprise Linux 8 and Red Hat Enterprise Linux 7 is systemd.

NOTE

To determine which system daemon is running, issue **ps -p 1 -o comm=**.

- For systemd:

```
$ ps -p 1 -o comm=

systemd
```



IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

2.2.2.1.1. Setting up and using the JBoss Web Server with systemd

Setting up the JBoss Web Server for systemd
As the root user, execute the **.postinstall.systemd** script:

```
# cd JWS_HOME/tomcat
# sh .postinstall.systemd
```

Controlling the JBoss Web Server with systemd
Systemd commands can only be issued by the root user.

- To enable the JBoss Web Server services to start at boot using systemd:

```
# systemctl enable jws5-tomcat.service
```

- To start the JBoss Web Server using systemd:

```
# systemctl start jws5-tomcat.service
```

NOTE

SECURITY_MANAGER variable is now deprecated for configurations based on the RHEL zips installations and this adds the following comment:

```
# SECURITY_MANAGER has been deprecated. To run tomcat under the Java Security Manager use:
  JAVA_OPTS="-Djava.security.manager -Djava.security.policy==\"$CATALINA_BASE/conf/"catalina.policy\""
```

- To stop the JBoss Web Server using systemd:

```
# systemctl stop jws5-tomcat.service
```

- To verify the status of the JBoss Web Server using systemd (the **status** operation can be executed by any user):

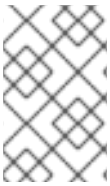
```
# systemctl status jws5-tomcat.service
```

For more information on using systemd on RHEL 7, see: RHEL 7 System Administrator's Guide: Managing System Services

For more information on using systemd on RHEL 8, see: RHEL 8 Configuring Basic System Settings: Managing system services with systemctl

2.2.2.2. Managing JBoss Web Server on a command line

2.2.2.2.1. Configuring the JBoss Web Server Installation



NOTE

The following configuration steps are performed by the **.postinstall.sysv** script and the **.postinstall.systemd** script described in Managing JBoss Web Server using a system daemon for .zip installations on Red Hat Enterprise Linux

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

- Setting the JAVA_HOME Environment Variable .
- Creating the tomcat user for simple and secure user management: Creating a Tomcat User .
- Grant the tomcat user access to the JBoss Web Server by moving the ownership of tomcat directory to the tomcat user.

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 (***JWS_HOME*/tomcat/bin**), create a file named **setenv.sh**, and insert the **JAVA_HOME** path definition.

For example: **export JAVA_HOME=/usr/lib/jvm/jre-1.8.0-openjdk.x86_64**

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 53 -r tomcat
```

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

```
# useradd -c "tomcat" -u 53 -g tomcat -s /sbin/nologin -r tomcat
```

Move the ownership of tomcat directory to the tomcat user

1. 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/
```

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

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

```
# chmod -R u+X tomcat/
```

2.2.2.2.2. Starting JBoss Web Server

Run the following command as the **tomcat** user:

```
$ sh JWS_HOME/tomcat/bin/startup.sh
```

2.2.2.2.3. Stopping JBoss Web Server

To stop Tomcat, run the following command as the **tomcat** user:

```
$ sh JWS_HOME/tomcat/bin/shutdown.sh
```

2.3. RPM INSTALLATION

Installing JBoss Web Server from RPM packages installs Tomcat as service, and installs its resources into absolute paths. The RPM installation option is available for Red Hat Enterprise Linux 7, and Red Hat Enterprise Linux 8.

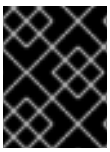
RPM installation packages for JBoss Web Server are available from Red Hat Subscription Management.

2.3.1. Attaching subscriptions to Red Hat Enterprise Linux

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.

For information on registering Red Hat Enterprise Linux, see the following procedures:

- The Subscription Manager for Red Hat Enterprise Linux 7
- The Subscription Manager for Red Hat Enterprise Linux 8

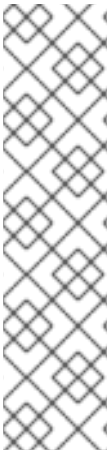


IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

Procedure

1. Log in to the Red Hat Subscription Manager .
2. Click on the **Systems** tab.
3. Click on the **Name** of the system to add the subscription to.
4. Change from the **Details** tab to the **Subscriptions** tab, then click **Attach Subscriptions**.
5. Select the check box beside the subscription to attach, then click **Attach Subscriptions**.



NOTE

To verify that a subscription provides the required CDN repositories:

1. Log in to: https://access.redhat.com/management/subscriptions.
2. Click the **Subscription Name**.
3. Under **Products Provided**, you require:
   - JBoss Enterprise Web Server.
   - Red Hat JBoss Core Services.

2.3.2. Installing JBoss Web Server from RPM packages using YUM

Prerequisites

- Install a Java Development Kit (JDK) .
- Ensure that the tomcatjss package is removed .

Procedure

1. On a command line, subscribe to the JBoss Web Server CDN repositories for your operating system version using **subscription-manager**:

```
# subscription-manager repos --enable <repository>
```

   - For Red Hat Enterprise Linux 7:
     - jws-5-for-rhel-7-server-rpms

- jb-coreservices-1-for-rhel-7-server-rpms

- For Red Hat Enterprise Linux 8:
  - jws-5-for-rhel-8-x86_64-rpms

2. Issue the following command as the root user to install JBoss Web Server:

```
# yum groupinstall jws5
```

IMPORTANT

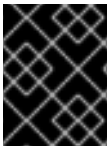
For RPM distributions, the JWS_HOME folder is **/opt/rh/jws5/root/usr/share**.

NOTE

- Although not recommended, instead of using the group install, you can install each of the packages and their dependencies individually.
- The Red Hat JBoss Core Services repositories above are required for the installation of JBoss Web Server except on RHEL 8 systems.
- The feature to enable NFS usage using Software Collection is enabled. For full instructions on this feature refer to the Packaging Guide, Using Software Collections over NFS.

2.3.3. Starting JBoss Web Server

This procedure demonstrates how you can start the JBoss Web Server.



IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

Procedure

- In a shell prompt as the root user, start the Tomcat service.
  - For Red Hat Enterprise Linux 7 or 8:

```
# systemctl start jws5-tomcat.service
```

NOTE

This is the only supported method of starting JBoss Web Server for an RPM installation.

- To verify that Tomcat is running, the output of the service **status** command should be reviewed. This can be executed as any user.
  - For Red Hat Enterprise Linux 7 or 8:

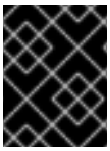
```
# systemctl status jws5-tomcat.service
```

NOTE

For complete instructions on installing and configuring HTTPD on RHEL 8, please see this link

2.3.4. Stopping JBoss Web Server

This procedure demonstrates how you can stop the JBoss Web Server.



IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

Procedure

- In a shell prompt as the root user, stop the Tomcat service.
  - For Red Hat Enterprise Linux 7 or 8:

```
# systemctl stop jws5-tomcat.service
```

- To verify that Tomcat is no longer running, the output of the service **status** command should be reviewed. This can be executed as any user.
  - For Red Hat Enterprise Linux 7 or 8:

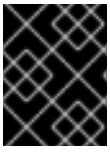
```
# systemctl status jws5-tomcat.service
```

NOTE

For complete instructions on installing and configuring HTTPD on RHEL 8, please see this link

2.3.5. Configuring JBoss Web Server Services to Start at Boot

Use the following commands to enable the JBoss Web Server services to start at boot.



IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

Procedure

- Depending on your Red Hat Enterprise Linux version, enter one of the following commands:
  - For Red Hat Enterprise Linux 7 or 8:

```
# systemctl enable jws5-tomcat.service
```

2.4. SELINUX POLICIES

2.4.1. SELinux Policy Information

The following table contains information about the SELinux policies provided in the jws5-tomcat-selinux packages.

Table 2.1. RPMs and Default SELinux Policies

Name	Port Information	Policy Information
jws5_tomcat	Four ports in **http_port_t** (TCP ports **8080**, **8005**, **8009**, and **8443**) to allow the tomcat process to use them.	The jws5_tomcat 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/opt/rh/jws5/lib/tomcat**, **/var/opt/rh/jws5/log/tomcat**, **/var/opt/rh/jws5/cache/tomcat** and **/var/opt/rh/jws5/run/tomcat.pid**.

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

2.4.2. SELinux policies for an RPM installation

SELinux policies for JBoss Web Server are provided by the **jws5-tomcat-selinux** package. These packages are available in the JWS channel.

To enable SELinux policies for JBoss Web Server 5.5, install the **jws5-tomcat-selinux** package.

2.4.3. SELinux policies for an archive installation

In this release, SELinux policies are provided in the archive packages. The SELinux security model is enforced by the kernel and ensures applications have limited access to resources such as file system locations and ports. This helps ensure that the errant processes (either compromised or poorly configured) are restricted and in some cases prevented from running.

The **.postinstall.selinux** file is included in the **tomcat** folder of **jws-5.5.0-application-server-*<platform>*-*<architecture>*.zip**. If required, you can run the **.postinstall.selinux** script.

To install the SELinux policies using archive:

1. Install the **selinux-policy-devel** package:

```
yum install -y selinux-policy-devel
```

2. Execute the **.postinstall.selinux** script:

```
cd <JWS_home>/tomcat/
sh .postinstall.selinux
```

3. Add access permissions to the required ports for JBoss Web Server. The JBoss Web Server has access to ports **8080**, **8009**, **8443** and **8005** on Red Hat Enterprise Linux systems.
   When additional ports are required for JBoss Web Server, use the **semanage** command to provide the necessary permissions, replacing the port number with the port required:

   ```
   semanage port -a -t http_port_t -p tcp <port>
   ```

4. Start Tomcat:

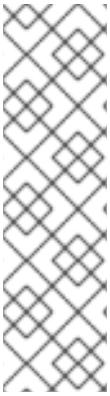
   ```
   <JWS_home>/tomcat/bin/startup.sh
   ```

5. Check the context of the running process expecting **jws5_tomcat**:

   ```
   ps -eo pid,user,label,args | grep jws5_tomcat | head -n1
   ```

6. To verify the contexts of the Tomcat directories, for example:

   ```
   ls -lZ <JWS_home>/tomcat/logs/
   ```



NOTE

By default, the SElinux policy provided is not active and the Tomcat processes run in the **unconfined_java_t** domain. This domain does not confine the processes, and it is recommended that you undertake the following security precautions if you chose not to enable the SElinux policy provided:

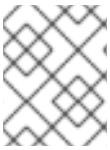
- Restrict file access for the **tomcat** user to only the files and directories that are necessary to the JBoss Web Server runtime.
- Do not run Tomcat as the **root** user.

CHAPTER 3. INSTALLING JBOSS WEB SERVER ON MICROSOFT WINDOWS

3.1. INSTALLING A JAVA DEVELOPMENT KIT (JDK)

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

For a list of supported configurations, see the Supporting Operating Systems and Configurations .



NOTE

For instructions on installing the IBM JDK, visit:
https://www.ibm.com/developerworks/java/jdk/

To install the Oracle Java Development Kit:

1. Download the Oracle JDK 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 installation.
3. Proceed as instructed in the installation window.

3.2. DOWNLOADING AND EXTRACTING JBOSS WEB SERVER

To install JBoss Web Server, download and extract the installation archive files.

1. Open a browser and log in to the Red Hat Customer Portal .
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:
   - The Red Hat JBoss Web Server 5.5 Application Server (**jws-5.5.0-application-server.zip**).
   - The Red Hat JBoss Web Server 5.5 Native Components for Windows Server (**jws-5.5.0-application-server-*<platform>*-*<architecture>*.zip**).
6. Unzip the downloaded archive files to your installation directory.

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

3.3. CONFIGURING THE JBOSS WEB SERVER INSTALLATION

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

- Setting Environment Variables
- Installing the Tomcat Service
- Configuring Folder Permissions for the JBoss Web Server Services

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** button 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 user is **SYSTEM** by default.

Installing the Tomcat Service

1. Open a command prompt with administrator privileges and change to the **bin** folder for your Tomcat version:

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

2. Install the Tomcat service with the following command:

```
call service.bat install
```

Configuring Folder Permissions for the 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.

3.4. STARTING JBOSS WEB SERVER

You can start the JBoss Web Server from a command prompt, or with the Computer Management tool.

Starting JBoss Web Server from a Command Prompt

1. Open a command prompt with administrator privileges.
2. Start the Tomcat service:

```
net start tomcat9
```

Starting JBoss Web Server from the Computer Management Tool

1. Go to **Start → Administrative Tools → Services**.
2. In the **Services** list, right-click the name of the service (**Tomcat9**) 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**.

3.5. STOPPING JBOSS WEB SERVER

You can stop the JBoss Web Server from a command prompt, or with the Computer Management tool.

Stopping JBoss Web Server from a Command Prompt

1. Open a command prompt with administrator privileges.
2. Stop the Tomcat service:

```
net stop tomcat9
```

Stopping JBoss Web Server from the Computer Management Tool

Stopping JBoss Web Server from the Computer Management Tool

1. Go to **Start → Administrative Tools → Services**.
2. In the **Services** list, right-click the name of the service (**Tomcat9**) and click **Stop**.

CHAPTER 4. CONFIGURING HIBERNATE FOR RED HAT JBOSS WEB SERVER

Hibernate ORM is an object-relational mapping framework that lets you connect JBoss Web Server to JDBC datasources.

4.1. INSTALLING HIBERNATE ORM

Complete the following procedure to install Hibernate ORM on all platforms that JBoss Web Server supports.

Prerequisites

Configure your project to use the JBoss Web Server Maven Repository, which is available to download as **jboss-web-server-5.5.0-maven-repository.zip**.

Procedure

1. Get the Hibernate JAR files from the JBoss Web Server Maven Repository.
2. Add the Hibernate JAR files to your deployment WAR file.

Reference

- Hibernate for JBoss Web Server Documentation

4.2. CONFIGURING JDBC CONNECTION POOLS

Tomcat provides a default connection pooling mechanism for JDBC datasources.

Procedure

1. Open your deployment's /**META-INF/context.xml** file for editing.
2. Modify the JDBC connection pools available to applications, as in the following example:

```
<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>
```

4.3. CONFIGURING HIBERNATE CONNECTION PROPERTIES

Configure Hibernate to use connections from the Tomcat pool. If you use the Hibernate API directly, use a similar configuration to **hibernate.cfg.xml**.

Procedure

1. Open your deployment's **/WEB-INF/classes/META-INF/persistence.xml** file for editing.
2. Configure how Hibernate consumes connections from the Tomcat, as in the following example:

```
<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>
```

4.4. ADDING JDBC DATA SOURCES

Configure Tomcat to consume JDBC datasources.

Procedure

1. Open your deployment's **/WEB-INF/web.xml** file for editing.
2. Configure JDBC datasources with the **resource-env-ref** element, as in the following example that uses a **jdbc/DsWebAppDB** datasource:

```
<?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>
```

CHAPTER 5. ENABLING HTTP/2 FOR THE RED HAT JBOSS WEB SERVER

The Hypertext Transfer Protocols are standard methods of transmitting data between applications (such as servers and browsers) over the internet.

HTTP/2 improves on HTTP/1.1 by providing enhancements such as:

- header compression - reducing the size of the header transmitted by omitting implied information, and
- multiple requests and responses over a single connection - using binary framing to break down response messages, as opposed to textual framing.

Using HTTP/2 with the Red Hat JBoss Web Server:

- **is supported** for encrypted connections over TLS (**h2**).
- **is not supported** for unencrypted connections over TCP (**h2c**).

Prerequisites

- Root user access (Red Hat Enterprise Linux systems)
- Red Hat JBoss Web Server 5.0 or higher
- The following operating system native libraries (provided by **jws-5.5.0-application-server-*<platform>*-*<architecture>*.zip** where available).
  RHEL-8 users needing to run JSSE+OpenSSL or APR, you will need to use Tomcat-Native for it to work properly. The file for Tomcat-Native can be found in the native archive directory. To install OpenSSL and APR, run the following commands:

```
# yum install openssl
```

```
# yum install apr
```

- A connector that supports the HTTP/2 protocol with SSL enabled. For JBoss Web Server 5.5, the connectors with HTTP/2 protocol support are:
  - The APR Native connector (APR)
  - The NIO connector with JSSE + OpenSSL (JSSE)
  - The NIO2 connector with JSSE + OpenSSL (JSSE)

Procedure

Enable HTTP/2 for a connector:

1. Add the HTTP/2 upgrade protocol (**<UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />**) to the connector in the server configuration ***JWS_HOME*/tomcat/conf/server.xml**.
   For example:

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
        maxThreads="150" SSLEnabled="true">
```

```
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
      <Certificate certificateKeystoreFile="/KeyStore.jks"
                   certificateKeystorePassword="changeit"
                   type="RSA" />
    </SSLHostConfig>
</Connector>
```

server.xml contains an example connector definition for the APR protocol with the upgrade protocol to HTTP/2:

```
<Connector port="8443"
           protocol="org.apache.coyote.http11.Http11AprProtocol"
           maxThreads="150" SSLEnabled="true" >
    <UpgradeProtocol className="org.apache.coyote.http2.Http2Protocol" />
    <SSLHostConfig>
      <Certificate certificateKeyFile="conf/localhost-rsa-key.pem"
                   certificateFile="conf/localhost-rsa-cert.pem"
                   certificateChainFile="conf/localhost-rsa-chain.pem"
                   type="RSA" />
    </SSLHostConfig>
</Connector>
```

2. Restart the Red Hat JBoss Web Server as the root user, to apply the changed configuration.

   a. For systemd (Red Hat Enterprise Linux 7) users:

   ```
   # systemctl restart jws5-tomcat.service
   ```

   b. For Red Hat Enterprise Linux users running Red Hat JBoss Web Server using **startup.sh**:

   ```
   # JWS_HOME/sbin/shudown.sh
   # JWS_HOME/sbin/startup.sh
   ```

   c. For Windows Server users:

   ```
   # net restart tomcat9
   ```

IMPORTANT

Red Hat Enterprise Linux 6 is no longer supported and subsequently was removed from the documentation.

Next Steps

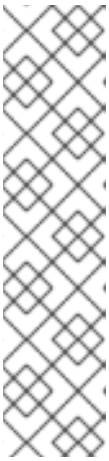
Verify that HTTP/2 is enabled by reviewing the Red Hat JBoss Web Server logs or by using the **curl** command:

- Check the console output log (***JWS_HOME*/tomcat/logs/catalina.out**) to verify that the "connector has been configured to support negotiation to [h2]":

```
$ cat JWS_HOME/tomcat/logs/catalina.out | grep 'h2'

06-Apr-2018 04:49:26.201 INFO [main]
```

```
org.apache.coyote.http11.AbstractHttp11Protocol.configureUpgradeProtocol The ["https-openssl-apr-8443"] connector has been configured to support negotiation to [h2] via ALPN
```

- Or verify using **curl** (for versions of **curl** that support **HTTP2**):



NOTE

To check **curl** for HTTP/2 support:

```
$ curl -V

curl 7.55.1 (x86_64-redhat-linux-gnu) ...
Release-Date: 2017-08-14
Protocols: dict file ftp ftps gopher http https ...
Features: AsynchDNS IDN IPv6 Largefile GSS-API Kerberos SPNEGO NTLM NTLM_WB SSL libz TLS-SRP HTTP2 UnixSockets HTTPS-proxy Metalink PSL
```

  - For example, when the HTTP/2 protocol is inactive:

```
$ curl -I http://<JBoss_Web_Server>:8080/

HTTP/1.1 200
...
```

  - But if the HTTP/2 protocol is active, **curl** returns:

```
$ curl -I https://<JBoss_Web_Server>:8443/

HTTP/2 200
...
```

    Where ***<JBoss_Web_Server>*** is the URI of the modified connector (such as **example.com**), and the port number is dependent on your configuration.

Additional Resources

- For additional information on using HTTP/2, see: Apache Tomcat 9 Configuration Reference: The HTTP Connector - HTTP/2 Support.
- For information on the HTTP/2 Upgrade Protocol and the supported attributes, see: Apache Tomcat 9 Configuration Reference: The HTTP2 Upgrade Protocol.
- The proposed internet standard for HTTP/2: IETF: RFC 7540 - Hypertext Transfer Protocol Version 2 (HTTP/2)

CHAPTER 6. VAULT FOR RED HAT JBOSS WEB SERVER

6.1. ABOUT PASSWORD VAULT IN RED HAT JBOSS WEB SERVER 5.5

Tomcat-vault is a **PicketLink vault** extension for Apache Tomcat that allows users to mask passwords and other sensitive strings, and store them in an encrypted Java keystore. Using the vault enables you to stop storing clear-text passwords in your Tomcat configuration files, because Tomcat can lookup passwords and other sensitive strings from a keystore using the vault.

IMPORTANT

For Using CRYPT with the Vault, refer Using CRYPT.

6.2. INSTALLING THE JBOSS WEB SERVER PASSWORD VAULT FROM .ZIP ARCHIVE

As tomcat password vault is pre-installed by the **jws-5.5.0-application-server.zip** file. The password vault can be used once configured and it is located at: ***JWS_HOME*/tomcat/lib/tomcat-vault.jar**.

6.3. INSTALLING THE JBOSS WEB SERVER PASSWORD VAULT ON RED HAT ENTERPRISE LINUX USING THE YUM PACKAGE MANAGER

If the JBoss Web Server has been installed from RPMs on Red Hat Enterprise Linux, you need to install the JBoss Web Server RPM for tomcat-vault.

Procedure

- Install the password vault as the root user by executing:

```
yum install jws5-tomcat-vault
```

6.4. ENABLING PASSWORD VAULT IN JBOSS WEB SERVER

In the following procedure, replace ***JWS_HOME*** with the path to your JBoss Web Server installation. Also, the paths below use / for directory separators.

Procedure

1. Stop Tomcat if it is running.
2. Edit ***JWS_HOME*/tomcat/conf/catalina.properties**, and add the following line:

```
org.apache.tomcat.util.digester.PROPERTY_SOURCE=org.apache.tomcat.vault.util.PropertySourceVault
```

6.5. CREATING A JAVA KEYSTORE IN JBOSS WEB SERVER

To use a password vault, you must first create a Java keystore.

IMPORTANT

The values in the procedure are examples only. Replace them with values specific to your environment.

For an explanation of the parameters, use the **keytool -genseckey -help** command.

Procedure

- Create a Java keystore using the **keytool -genseckey** command:

```
$ keytool -genseckey -keystore JWS_HOME/tomcat/vault.keystore -alias my_vault  -
storetype jceks -keyalg AES -keysize 128 -storepass <vault_password> -keypass
<vault_password> -validity 730
```

IMPORTANT

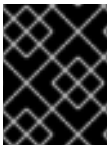
At this time, keystore type **PKCS12** is not supported by tomcat-vault. Only keystore type **JCEKS** is supported.

Additionally, the following keystore algorithms must have the following keysize:

- AES: **-keysize 128**
- DES: **-keysize 56**
- DESede: **-keysize 168**

6.6. INITIALIZING PASSWORD VAULT

6.6.1. Initializing password vault for Apache Tomcat interactively



IMPORTANT

The values below are examples only. Replace them with values appropriate for your environment.

Procedure

- Initialize password vault using the **tomcat-vault.sh** script:

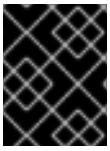
```
# JWS_HOME/tomcat/bin/tomcat-vault.sh

WARNING JBOSS_HOME may be pointing to a different installation - unpredictable results
may occur.

=========================================================================


 JBoss Vault

 JBOSS_HOME: JWS_HOME/tomcat
```

```
 JAVA: java

================================================================================


**********************************
****  JBoss Vault  ***************
**********************************
Please enter a Digit::
0: Start Interactive Session
1: Remove Interactive Session
2: Exit

0

Starting an interactive session
Enter directory to store encrypted files: JWS_HOME/tomcat/
Enter Keystore URL: JWS_HOME/tomcat/vault.keystore
Enter Keystore password: <vault_password>
Enter Keystore password again: <vault_password>
Values match
Enter 8 character salt: 1234abcd
Enter iteration count as a number (Eg: 44): 120
Enter Keystore Alias: my_vault
Initializing Vault
Jun 16, 2018 10:24:27 AM org.apache.tomcat.vault.security.vault.PicketBoxSecurityVault init
INFO: PBOX000361: Default Security Vault Implementation Initialized and Ready
Vault Configuration in tomcat properties file:
********************************************
...
KEYSTORE_URL=JWS_HOME/tomcat/vault.keystore
KEYSTORE_PASSWORD=MASK-3CuP21KMHn7G6iH/A3YpM/
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=JWS_HOME/tomcat/
...
********************************************
Vault is initialized and ready for use
Handshake with Vault complete
Please enter a Digit::
0: Store a secured attribute
1: Check whether a secured attribute exists
2: Exit

2
```

Note the output for the Tomcat properties file, as you will need this to configure Tomcat to use the vault.

6.6.2. Initializing the Vault for Apache Tomcat non-interactively (silent setup)

The Vault for Apache Tomcat can be created non-interactively by providing the required input as arguments to the **tomcat-vault.sh** script. The **vault.properties** file is also created as output of the **tomcat-vault.sh** script when the **-g, --generate-config** option is used.



IMPORTANT

The values below are examples only. Replace them with values appropriate for your environment.

Procedure

- Initialize password vault using the **tomcat-vault.sh** script:

```
$ JWS_HOME/tomcat/bin/tomcat-vault.sh \
 --keystore JWS_HOME/tomcat/vault.keystore \
 --keystore-password <vault_password> \
 --alias my_vault \
 --enc-dir JWS_HOME/tomcat/ \
 --iteration 120 \
 --salt 1234abcd \
 --generate-config JWS_HOME/tomcat/conf/vault.properties
```

6.7. CONFIGURING TOMCAT TO USE THE PASSWORD VAULT

Prerequisites

- Password vault for Tomcat is initialized.
  For information about initializing password vault for Tomcat, see Initializing password vault for Apache Tomcat interactively

Procedure

- In ***JWS_HOME*/tomcat/conf/**, create a file named **vault.properties** containing the vault configuration produced when initializing the vault.
  The values provided below use the example vault initialized in procedure Initializing password vault for Apache Tomcat interactively

NOTE

For **KEYSTORE_PASSWORD**, you must use the masked value that was generated when initializing the vault.

```
KEYSTORE_URL=JWS_HOME/tomcat/vault.keystore
KEYSTORE_PASSWORD=MASK-3CuP21KMHn7G6iH/A3YpM/
KEYSTORE_ALIAS=my_vault
SALT=1234abcd
ITERATION_COUNT=120
ENC_FILE_DIR=JWS_HOME/tomcat/
```

6.8. EXTERNAL PASSWORD VAULT CONFIGURATION

The **vault.properties** file for the **tomcat-vault** can be stored outside of ***JWS_HOME*/tomcat/conf/** in a ***CATALINA_BASE*/conf/** directory (if set).

To set the *CATALINA_BASE* directory, follow the instructions in the section **Advanced Configuration - Multiple Tomcat Instances** in the Running The Apache Tomcat 9.0 Servlet/JSP Container document found on the Apache Tomcat Website.

NOTE

The default location for *CATALINA_BASE* is ***JWS_HOME*/tomcat/** also known as *CATALINA_HOME*.

Additional Resources

For more information on setting *CATALINA_BASE*, see:

- Apache Tomcat 9: Introduction - Directories and Files
- Running The Apache Tomcat 9.0 Servlet/JSP Container: Advanced Configuration - Multiple Tomcat Instances

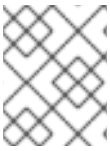
6.9. STORING A SENSITIVE STRING IN THE PASSWORD VAULT

The vault script used in the previous steps is also used to store sensitive strings in the password vault. The script can be run interactively or non-interactively.

When adding a string to a password vault, the sensitive string needs a name that it will be referred by. For a password vault, this name is called an **attribute name**, and the password itself is called a **secured attribute**.

The example below demonstrates using the vault script non-interactively to store a password. It uses the vault that was initialized in the previous steps, and stores the sensitive string **P@SSW0#D** with the attribute name **manager_password**.

```
$ JWS_HOME/tomcat/bin/tomcat-vault.sh --keystore JWS_HOME/tomcat/vault.keystore --keystore-password <vault_password> --alias my_vault --enc-dir JWS_HOME/tomcat --iteration 120 --salt 1234abcd --vault-block my_block --attribute manager_password --sec-attr P@SSW0#D
```



NOTE

You can optionally specify a vault block to store the password in. If you don't specify a block, one will be automatically created for you. In the above example, my_block is used.

6.10. USING A STORED SENSITIVE STRING IN YOUR TOMCAT CONFIGURATION

After storing a sensitive string in the password vault, you can refer to it in your configuration files by entering the stored string's attribute as **${VAULT::block_name::attribute_name::}**.

For example, to use the password stored in the previous steps, replace:

```
<user username="manager" password="P@SSW0#D" roles="manager-gui"/>
```

with:

```
<user username="manager" password="${VAULT::my_block::manager_password::}"
roles="manager-gui"/>
```

As a result, only a reference to the password is visible in the Tomcat configuration file, and the actual password is only stored in the password vault.

CHAPTER 7. CONFIGURING SSI FILTER

7.1. CONFIGURING THE SSI FILTER

SSI directives do not work if you try to configure SSI filter like you did in the previous versions.

Procedure

For the SSI Filter configuration to work correctly, uncomment the following block in the **conf/web.xml** file:

```
<mime-mapping>
    <extension>shtml</extension>
    <mime-type>text/x-server-parsed-html</mime-type>
</mime-mapping>
```

APPENDIX A. JAVA IPV4/IPV6 PROPERTIES

Configuring Java properties

In Java there are 2 properties that are used to configure IPv4 and IPv6. These are **java.net.preferIPv4Stack** and **java.net.preferIPv6Addresses**.

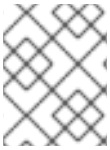
java.net.preferIPv4Stack (default: false)

If IPv6 is available then the underlying native socket, by default, is an IPv6 socket. This socket lets applications connect and accept connections from IPv4 and IPv6 hosts. If application use only IPv4 sockets, then set this property to **true**. However, it will not be possible for the application to communicate with IPv6 only hosts.

java.net.preferIPv6Addresses (default: false)

If a host has both IPv4 and IPv6 addresses, and IPv6 is available, then the default behavior is to use IPv4 addresses over IPv6. This allows backward compatibility. If applications that depend on an IPv4 address representation, for example: 192.168.1.1. Then, set this property to **true** to change the preference and use IPv6 addresses over IPv4 where possible.

To pass these properties to Tomcat, set **CATALINA_OPTS** in the ***JWS_HOME*/tomcat/bin/setenv.*** file.



NOTE

If the ***JWS_HOME*/tomcat/bin/setenv.sh** or ***JWS_HOME*/tomcat/bin/setenv.bat** file does not exist, then you need to create one.

On Linux:

export "CATALINA_OPTS=-Djava.net.preferIPv4Stack=*YOUR_VALUE* -Djava.net.preferIPv6Addresses=*YOUR_VALUE*"

On Windows:

set "CATALINA_OPTS=-Djava.net.preferIPv4Stack=*YOUR_VALUE* -Djava.net.preferIPv6Addresses=*YOUR_VALUE*"

Configuring Tomcat bindings

The Tomcat bindings can be set in ***JWS_HOME*/tomcat/conf/server.xml** with the IPv6 address:

- Specify the Tomcat binding address:
  <Server … address="*TOMCAT_BINDING_ADDRESS*">
- Specify the HTTP connector address:
  <Connector protocol="HTTP/1.1" … address="*HTTP_CONNECTOR_ADDRESS*">
- Specify the AJP connector address:
  <Connector protocol="AJP/1.3" … address="*AJP_CONNECTOR_ADDRESS*">