# Red Hat JBoss Core Services 2.4.37

## Apache HTTP Server Installation Guide

For use with Red Hat JBoss middleware products.

Last Updated: 2023-01-04

# Red Hat JBoss Core Services 2.4.37 Apache HTTP Server Installation Guide

For use with Red Hat JBoss middleware products.

## Legal Notice

## Abstract

Install, upgrade, and configure the Red Hat JBoss Core Services Apache HTTP Server on supported operating systems.

# Table of Contents

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION 4

CHAPTER 1. INTRODUCTION TO JBOSS CORE SERVICES APACHE HTTP SERVER INSTALLATION 5
1.1. JBOSS CORE SERVICES APACHE HTTP SERVER 5
1.2. SUPPORTED OPERATING SYSTEMS FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER 5
1.3. INSTALLATION METHODS FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER 6
1.4. UPGRADING TO THE JBOSS CORE SERVICES APACHE HTTP SERVER 2.4.37 6
1.5. KEY DIFFERENCES BETWEEN RED HAT ENTERPRISE LINUX 7 AND RED HAT ENTERPRISE LINUX 8 7
1.6. ADDITIONAL RESOURCES (OR NEXT STEPS) 7

CHAPTER 2. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON RHEL FROM ARCHIVE FILES 8
2.1. DOWNLOADING AND EXTRACTING THE APACHE HTTP SERVER ARCHIVE FILE ON RHEL 8
2.2. APACHE HTTP SERVER CONFIGURATION FOR MANAGING ARCHIVE INSTALLATIONS FROM THE COMMAND LINE 9
2.2.1. Creating an Apache user 10
2.2.2. Disabling or enabling SSL support 10
2.2.3. Running the Apache HTTP Server post-installation script 11
2.3. STARTING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM AN ARCHIVE FILE 11
2.4. STOPPING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM AN ARCHIVE FILE 11
2.5. RUNNING THE APACHE HTTP SERVER FROM THE COMMAND LINE WITHOUT ROOT ACCESS 12
2.6. MANAGING APACHE HTTP SERVER BY USING SYSTEMD WHEN INSTALLED FROM AN ARCHIVE FILE 12
2.7. SELINUX POLICIES FOR THE APACHE HTTP SERVER 14
2.7.1. SELinux policy information 14
2.7.2. Installing SELinux policies for an Apache HTTP Server archive installation 14

CHAPTER 3. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON RHEL FROM RPM PACKAGES 16
3.1. ATTACHING SUBSCRIPTIONS TO RED HAT ENTERPRISE LINUX 16
3.2. INSTALLING THE APACHE HTTP SERVER FROM RPM PACKAGES BY USING YUM 17
3.3. USING MOD_JK AND MOD_CLUSTER WITH RHEL 8 17
3.4. CONFIGURING THE APACHE HTTP SERVER INSTALLATION WHEN INSTALLED FROM RPMS 18
3.5. STARTING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM RPMS 19
3.6. STOPPING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM RPMS 19
3.7. CONFIGURING THE APACHE HTTP SERVER SERVICE TO START AT SYSTEM STARTUP 20
3.8. SELINUX POLICIES FOR THE APACHE HTTP SERVER 20
3.8.1. SELinux policy information 20
3.8.2. Enabling SELinux policies for an Apache HTTP Server RPM installation 20

CHAPTER 4. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON MICROSOFT WINDOWS 22
4.1. DOWNLOADING AND EXTRACTING THE APACHE HTTP SERVER ARCHIVE FILE ON MICROSOFT WINDOWS 22
4.2. APACHE HTTP SERVER CONFIGURATION ON MICROSOFT WINDOWS 23
4.2.1. Running the Apache HTTP Server post-installation script on Microsoft Windows 23
4.2.2. Installing the Apache HTTP Server service 23
4.2.3. Configuring folder permissions for the Apache HTTP Server service 24
4.2.4. Disabling or enabling SSL support 24
4.3. STARTING THE APACHE HTTP SERVER ON MICROSOFT WINDOWS 25

4.4. STOPPING THE APACHE HTTP SERVER ON MICROSOFT WINDOWS 25

**CHAPTER 5. ENABLING HTTP/2 FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER . . . . . . . . . . . 27**
5.1. PREREQUISITES 27
5.2. ENABLING HTTP/2 FOR THE APACHE HTTP SERVER 27
5.3. VIEWING APACHE HTTP SERVER LOGS TO VERIFY THAT HTTP/2 IS ENABLED 29
5.4. USING THE CURL COMMAND TO VERIFY THAT HTTP/2 IS ENABLED 30
5.5. ADDITIONAL RESOURCES (OR NEXT STEPS) 31

# PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our technical content and encourage you to tell us what you think. If you’d like to add comments, provide insights, correct a typo, or even ask a question, you can do so directly in the documentation.

NOTE

You must have a Red Hat account and be logged in to the customer portal.

To submit documentation feedback from the customer portal, do the following:

1. Select the **Multi-page HTML** format.
2. Click the **Feedback** button at the top-right of the document.
3. Highlight the section of text where you want to provide feedback.
4. Click the **Add Feedback** dialog next to your highlighted text.
5. Enter your feedback in the text box on the right of the page and then click **Submit**.

We automatically create a tracking issue each time you submit feedback. Open the link that is displayed after you click **Submit** and start watching the issue or add more comments.

Thank you for the valuable feedback.

# CHAPTER 1. INTRODUCTION TO JBOSS CORE SERVICES APACHE HTTP SERVER INSTALLATION

Red Hat JBoss Core Services is a set of supplementary software for Red Hat JBoss middleware products. Red Hat JBoss Core Services provides supplementary software, such as the Apache HTTP Server, that is common to multiple JBoss middleware products. This supplementary software is packaged under Red Hat JBoss Core Services to allow for faster distribution of updates, and for a more consistent update experience.

For a full list of components that are supported by Red Hat JBoss Core Services, see the Core Services Apache HTTP Server Component Details web page.

NOTE

Before you attempt to access the Core Services Apache HTTP Server Component Details web page, you must ensure that you have an active Red Hat subscription, and you are logged in to the Red Hat Customer Portal.

## 1.1. JBOSS CORE SERVICES APACHE HTTP SERVER

The Red Hat JBoss Core Services Apache HTTP Server is used in multiple Red Hat JBoss middleware products. The Apache HTTP Server processes requests that web clients send over the Hypertext Transfer Protocol (HTTP).

In older JBoss product releases, the Apache HTTP Server was distributed with each JBoss product separately. Starting from the following product versions, each JBoss middleware product uses the JBoss Core Services distribution of the Apache HTTP Server:

- Red Hat JBoss Enterprise Application Platform (JBoss EAP) 7.0 and later versions
- Red Hat JBoss Web Server 3.1 and later versions

IMPORTANT

Both JBoss Core Services and Red Hat Enterprise Linux provide separate distributions of the Apache HTTP Server.

Consider the following differences between the Apache HTTP Server distributions that are available with JBoss Core Services (JBCS) and Red Hat Enterprise Linux (RHEL):

- You can install the JBCS Apache HTTP Server from an archive file or RPM package. You can only install the RHEL Apache HTTP Server from an RPM package.
- The JBCS Apache HTTP Server provides the **mod_security** module, the **mod_proxy_uwsgi** module, and the loadbalancing modules **mod_jk** and **mod_cluster**.
- The JBCS Apache HTTP Server does not provide or support the **mod_php** module. The RHEL Apache HTTP Server supports the **mod_php** module.

## 1.2. SUPPORTED OPERATING SYSTEMS FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER

The JBoss Core Services Apache HTTP Server supports different versions of the Red Hat Enterprise Linux and Microsoft Windows operating systems.

### Additional resources

- Core Services HTTP Server Supported Configurations web page.

## 1.3. INSTALLATION METHODS FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER

You can install the JBoss Core Services Apache HTTP Server on supported Red Hat Enterprise Linux and Microsoft Windows systems by using archive installation files that are available for each platform. You can also install the JBoss Core Services Apache HTTP Server on supported Red Hat Enterprise Linux systems by using RPM packages.

## 1.4. UPGRADING TO THE JBOSS CORE SERVICES APACHE HTTP SERVER 2.4.37

If you have previously installed an earlier version of the JBoss Core Services Apache HTTP Server from an archive file, you can upgrade to the JBoss Core Services Apache HTTP Server 2.4.37 release.

The upgrade process includes the following steps:

1. Installing the Apache HTTP Server 2.4.37.

2. Setting up the Apache HTTP Server 2.4.37.

3. Removing an earlier version of Apache HTTP Server.

### Prerequisites

- You have administrative access on Windows Server.

- You are using a system where the JBoss Core Services Apache HTTP Server 2.4.29 or earlier was installed from a .zip archive.

### Procedure

1. Shut down any running instances of the Apache HTTP Server 2.4.29.

2. Back up the Apache HTTP Server 2.4.29 installation and configuration files.

3. Install the Apache HTTP Server 2.4.37 by using the .zip installation method for the current system. For more information see Additional Resources at the end of this section.

4. Migrate your configuration from the Apache HTTP Server version 2.4.29 to version 2.4.37.

   **NOTE**

   The JBoss Core Services configuration files might have changed since the Apache HTTP Server 2.4.29 release. Update the 2.4.37 version configuration files rather than overwrite them with the configuration files from a different version, such as the Apache HTTP Server 2.4.29.

5. Remove the Apache HTTP Server 2.4.29 root directory.

### Additional resources

- Installing the JBoss Core Services Apache HTTP Server on Microsoft Windows

## 1.5. KEY DIFFERENCES BETWEEN RED HAT ENTERPRISE LINUX 7 AND RED HAT ENTERPRISE LINUX 8

This section provides an overview of some of the key changes introduced in Red Hat Enterprise Linux 8.

**Removed security functionality**

All-numeric user and group names are deprecated in Red Hat Enterprise Linux 7 and their support is completely removed in Red Hat Enterprise Linux 8.

**Memory management**

In Red Hat Enterprise Linux 7, the existing memory bus has capacity for 48/46 bit of virtual/physical memory addressing, and the Linux kernel implements 4 levels of page tables to manage these virtual addresses to physical addresses. With the extended address range, the memory management in Red Hat Enterprise Linux 8 supports the implementation of 5-level page tables, to allow handling of the expanded address range. In Red Hat Enterprise Linux 8, support for 5-level page tables is disabled by default, even if the system supports this feature.

**XFS supports**

Red Hat Enterprise Linux 7 can mount XFS file systems with shared copy-on-write data extents only in the read-only mode. In Red Hat Enterprise Linux 8, the XFS file system supports shared copy-on-write data extent functionality. This feature enables two or more files to share a common set of data blocks.

**NFS configuration**

In Red Hat Enterprise Linux 7, the NFS configuration is located in the **/etc/sysconfig/nfs** file. In Red Hat Enterprise Linux 8, the NFS configuration is located in the **/etc/nfs.conf** file.

> NOTE
>
> For more information about the differences between Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8, see Considerations in adopting RHEL 8 .

## 1.6. ADDITIONAL RESOURCES (OR NEXT STEPS)

- Installing the JBoss Core Services Apache HTTP Server on RHEL from archive files
- Installing the JBoss Core Services Apache HTTP Server on RHEL from RPM packages
- Installing the JBoss Core Services Apache HTTP Server on Microsoft Windows

# CHAPTER 2. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON RHEL FROM ARCHIVE FILES

You can install the JBoss Core Services Apache HTTP Server on Red Hat Enterprise Linux from archive files or RPM packages. If you want to install the Apache HTTP Server from archive files, you can download and extract the Apache HTTP Server from the Software Downloads page on the Red Hat Customer Portal. You must install the base archive file for the original 2.4.37 release. You can also install the latest service pack release, if any.

When you install the Apache HTTP Server from an archive file, you can manage the product in different ways. For example, you can use a system daemon at system startup or manage the Apache HTTP Server from a command line.

NOTE

The steps to download the Apache HTTP Server archive file on Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8 are different.

## 2.1. DOWNLOADING AND EXTRACTING THE APACHE HTTP SERVER ARCHIVE FILE ON RHEL

You can download the Apache HTTP Server archive files from the Software Downloads page on the Red Hat Customer Portal. Depending on the Red Hat Enterprise Linux (RHEL) version that you are using, the steps to download the archive file are slightly different.

Consider the following guidelines:

- If you are using RHEL 7, you must download the archive file for the original JBoss Core Services Apache HTTP Server 2.4.37 release from the **Releases** tab on the Software Downloads page. You can also download the latest service pack release, if any, from the **Security Advisories** tab on the Software Downloads page.

- If you are using RHEL 8, you must download the archive file for the original JBoss Core Services Apache HTTP Server 2.4.37 release from the **Security Advisories** tab on the Software Downloads page. You can also download the latest service pack release, if any, from the **Security Advisories** tab.

NOTE

You can install the archive file with non-root privileges, provided that you have write access to the intended installation directory.

### Prerequisites

- You have installed the **elinks**, **krb5-workstation**, and **mailcap** packages.
  If you want to install these packages, enter the following command as the root user:

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

### Procedure

1. Open a browser and log in to the Software Downloads page on the Red Hat Customer Portal.

2. From the **Product** drop-down menu, select **Apache HTTP Server**.

3. From the **Version** drop-down menu, select the correct JBoss Core Services version.

4. Depending on the RHEL version that you are using, perform either of the following steps:

   a. If you are using RHEL 7, on the **Releases** tab, click **Download** next to the Red Hat JBoss Core Services Apache HTTP Server archive file that matches the platform and architecture for your system.

   b. If you are using RHEL 8, click the **Security Advisories** tab. Then click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.37 Patch 06 for RHEL 8 x86_64** file.

   > NOTE
   >
   > The **Red Hat JBoss Core Services Apache HTTP Server 2.4.37 Patch 06 for RHEL 8 x86_64** file is the base archive file for installing the Apache HTTP Server on RHEL 8.

5. Extract the downloaded archive file to your installation directory.

   > NOTE
   >
   > On Red Hat Enterprise Linux systems, install the Apache HTTP Server in the **/opt/** directory.

   The **jbcs-httpd24-2.4/httpd** directory is created when you extract the archive. This directory is the top-level directory for the Apache HTTP Server. This document refers to the **jbcs-httpd24-2.4/httpd** directory as ***HTTPD_HOME***.

6. To install the latest service pack release, if any, perform the following steps:

   a. On the Software Downloads page, click the **Security Advisories** tab.

   b. On the **Security Advisories** tab, click **Download** next to the latest Red Hat JBoss Core Services Apache HTTP Server Patch archive file that matches the platform and architecture for your system.
   For example, if you want to install the Service Pack 10 release of the Apache HTTP Server 2.4.37 on Red Hat Enterprise Linux 7, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.37 Patch 10 for RHEL 7 x86_64** file.

   > NOTE
   >
   > Service pack releases are cumulative. By downloading the latest service pack release, you also install any previous service pack releases automatically.

## 2.2. APACHE HTTP SERVER CONFIGURATION FOR MANAGING ARCHIVE INSTALLATIONS FROM THE COMMAND LINE

When you install the JBoss Core Services Apache HTTP Server from an archive file on Red Hat Enterprise Linux, you can start and stop the Apache HTTP Server directly from the command line. Before you can run the Apache HTTP Server from the command line, you must perform the following series of configuration tasks:

- Create an Apache user
- Disable or enable SSL support
- Run the Apache HTTP Server post-installation script

### 2.2.1. Creating an Apache user

Before you run the Apache HTTP Server from the command line for the first time, you must create the **apache** user and its parent group. You must also assign ownership of the Apache directories to the **apache** user, so that the user can run the Apache HTTP Server.

NOTE

You must perform all steps in this procedure as the root user.

#### Prerequisites

- You have installed the Apache HTTP Server from an archive file .

#### Procedure

1. On a command line, go to the ***HTTPD_HOME*** directory.
2. To create the **apache** user group, enter the following command:

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

3. To create the **apache** user in the **apache** user group, enter the following command:

```
# /usr/sbin/useradd -c "Apache" -u 48 -g apache -s /sbin/nologin -r apache
```

4. To assign ownership of the Apache directories to the **apache** user, enter the following command:

```
# chown -R apache:apache *
```

#### Verification

1. To verify that the **apache** user is the owner of the directory, enter the following command:

```
# ls -l
```

### 2.2.2. Disabling or enabling SSL support

Before you run the Apache HTTP Server, you can choose to disable or enable SSL support by renaming the SSL configuration file. The Apache HTTP Server supports SSL by default.

#### Procedure

1. Go to the ***HTTPD_HOME*/conf.d/** directory.
2. To enable or disable SSL, perform either of the following steps:

- If you want to disable SSL, rename **ssl.conf** to **ssl.conf.disabled**.
- If you want to re-enable SSL, rename **ssl.conf.disabled** to **ssl.conf**.

### 2.2.3. Running the Apache HTTP Server post-installation script

Before you run the Apache HTTP Server from the command line for the first time, you must run the Apache HTTP Server post-installation script.

**Procedure**

1. On a command line, go to the ***HTTPD_HOME*** directory.
2. Enter the following command:

```
./.postinstall
```

## 2.3. STARTING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM AN ARCHIVE FILE

When you install JBoss Core Services Apache HTTP Server from an archive file on Red Hat Enterprise Linux, you can start the Apache HTTP Server directly from the command line.

**Prerequisites**

- You have created an **apache** user.
- You have disabled or re-enabled SSL support.
- You have run the Apache HTTP Server post-installation script .

**Procedure**

1. On a command line, go to the ***HTTPD_HOME*/sbin/** directory.
2. Enter the following command as the root user:

```
./apachectl start
```

## 2.4. STOPPING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM AN ARCHIVE FILE

When you install JBoss Core Services Apache HTTP Server from an archive file on Red Hat Enterprise Linux, you can stop a running instance of the Apache HTTP Server directly from the command line.

**Prerequisites**

- You have started the Apache HTTP Server.

**Procedure**

1. On a command line, go to the ***HTTPD_HOME*/sbin/** directory.

2. Enter the following command as the root user:

```
./apachectl stop
```

## 2.5. RUNNING THE APACHE HTTP SERVER FROM THE COMMAND LINE WITHOUT ROOT ACCESS

When you install JBoss Core Services Apache HTTP Server from an archive file on Red Hat Enterprise Linux, you can start the Apache HTTP Server directly from the command line as a user without root access, such as the **apache** user.

### Procedure

1. Stop all instances of the Apache HTTP Server:

2. Set the **http** listen port to higher than 1024 in

```
Listen 2080
ServerName <hostname>:2080
```

3. Set the **https** listen port to higher than 1024 in

```
Listen 2443
```

4. Change the ownership of the

5. Change the ownership of the **run**

6. Verify that **httpd** is running under the **apache** user only rather than the **root** and **apache** users:

```
$ ps -eo euser,egroup,comm | grep httpd
```

This command produces the following type of output:

```
apache   apache   httpd
apache   apache   httpd
apache   apache   httpd
...
```

> **IMPORTANT**
>
> Limit the file permissions of the **apache** user . This helps to prevent the following scenarios:
>
> - Unauthorized access or modification of files and directories by website users
> - Unwanted changes to the Apache HTTP Server configuration files

## 2.6. MANAGING APACHE HTTP SERVER BY USING SYSTEMD WHEN INSTALLED FROM AN ARCHIVE FILE

When you install the Apache HTTP Server from an archive file on Red Hat Enterprise Linux, you can use a system daemon to perform management tasks. Using the Apache HTTP Server with a system daemon provides a method of starting the Apache HTTP Server services at system startup. The system daemon also provides start, stop and status check functions.

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

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

### Prerequisites

- You have installed the Apache HTTP Server from an archive file .

### Procedure

1. To determine which system daemon is running, enter the following command:

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

   If **systemd** is running, the following output is displayed:

   ```
   systemd
   ```

2. To set up the Apache HTTP Server for **systemd**, run the **.postinstall.systemd** script as the root user:

   ```
   # cd HTTPD_HOME
   # sh httpd/.postinstall.systemd
   ```

3. To control the Apache HTTP Server with **systemd**, you can perform any of the following steps as the root user:

   - To enable the Apache HTTP Server services to start at system startup by using **systemd**:

     ```
     # systemctl enable jbcs-httpd24-httpd.service
     ```

   - To start the Apache HTTP Server by using **systemd**:

     ```
     # systemctl start jbcs-httpd24-httpd.service
     ```

   - To stop the Apache HTTP Server by using **systemd**:

     ```
     # systemctl stop jbcs-httpd24-httpd.service
     ```

   - To verify the status of the Apache HTTP Server by using **systemd**:

     ```
     # systemctl status jbcs-httpd24-httpd.service
     ```

NOTE

Any user can run the **status** operation.

IMPORTANT

After you run these commands, you can run the following command to revert changes affected by **.postinstall.sysv** or **.postinstall.systemd**:

```
# cd HTTPD_HOME
# sh httpd/.postinstall.services.cleanup
```

Additional resources

- For more information about using **systemd** on RHEL 7, see RHEL 7 System Administrator's Guide: Managing System Services.
- For more information about using **systemd** on RHEL 8, see RHEL 8 Configuring Basic System Settings: Managing system services with systemctl.

## 2.7. SELINUX POLICIES FOR THE APACHE HTTP SERVER

You can use Security-Enhanced Linux (SELinux) policies to define access controls for the Apache HTTP Server. These policies are a set of rules that determine access rights to the product.

### 2.7.1. SELinux policy information

The SELinux security model is enforced by the kernel and ensures that applications have limited access to resources such as file system locations and ports. SELinux policies ensure that any errant processes that are compromised or poorly configured are restricted or prevented from running.

The **jbcs-httpd24-httpd-selinux** packages in your Apache HTTP Server installation provide a **mod_cluster** policy. The following table contains information about the supplied SELinux 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. |

Additional resources

- For more information about using SELinux on Red Hat Enterprise Linux 7, see the RHEL 7 SELinux User's and Administrator's Guide.
- For more information about using SELinux on Red Hat Enterprise Linux 8, see the RHEL8 Using SELinux guide.

### 2.7.2. Installing SELinux policies for an Apache HTTP Server archive installation

In this release, the archive packages provide SELinux policies. The **.postinstall.selinux** file is included in the root Apache HTTP Server folder. If required, you can run the **.postinstall.selinux** script.

### IMPORTANT

By default, the SELinux policy that the Apache HTTP Server provides is not active and the Apache HTTP Server processes run in the **unconfined_t** domain. This domain does not confine the processes. If you choose not to enable the SELinux policy that is provided, restrict file access for the **apache** user, so that the **apache** user only has access to the files and directories that are necessary for the Apache HTTP Server runtime.

### Procedure

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

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

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

```
cd <httpd_home>
sh .postinstall.selinux
```

3. Make and install the SELinux module:

```
cd <httpd_home>/selinux/
make -f /usr/share/selinux/devel/Makefile
semodule -i jbcs-httpd24-httpd.pp
```

4. Apply the SELinux contexts for the Apache HTTP Server:

```
restorecon -r <httpd_home>
```

5. Add access permissions to the required ports for the Apache HTTP Server:

```
semanage port -a -t http_port_t -p tcp 6666
semanage port -a -t http_port_t -p udp 23364
```

6. Start the Apache HTTP Server service:

```
<httpd_home>/sbin/apachectl start
```

7. Check the context of the running process expecting **httpd_t**:

```
$ ps -eZ | grep httpd | head -n1

unconfined_u:unconfined_r:httpd_t:s0-s0:c0.c1023 2864 ? 00:00:00 httpd
```

8. Verify the contexts of the httpd directories. For example:

```
ls -lZ <httpd_home>/logs/
```

# CHAPTER 3. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON RHEL FROM RPM PACKAGES

You can install the JBoss Core Services Apache HTTP Server on Red Hat Enterprise Linux from archive files or RPM packages. Installing the Apache HTTP Server from RPM packages installs the Apache HTTP Server as a service.

RPM installation packages for JBoss Core Services Apache HTTP Server are available from Red Hat Subscription Management. The RPM installation option is available for Red Hat Enterprise Linux 7 and Red Hat Enterprise Linux 8.

IMPORTANT

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

## 3.1. ATTACHING SUBSCRIPTIONS TO RED HAT ENTERPRISE LINUX

Before you download and install the RPM packages for the Apache HTTP Server, you must register your system with Red Hat Subscription Management, and subscribe to the respective Content Delivery Network (CDN) repositories. You can subsequently perform some verification steps to ensure that a subscription provides the required CDN repositories.

### Procedure

1. Log in to the Red Hat Subscription Management web page.
2. Click the **Systems** tab.
3. Click the **Name** of the system that you want to add the subscription to.
4. Change from the **Details** tab to the **Subscriptions** tab, and then click **Attach Subscriptions**.
5. Select the check box next to the subscription that you want to attach, and then click **Attach Subscriptions**.

### Verification

1. Log in to the Red Hat Subscriptions web page.
2. In the **Subscription Name** column, click the subscription that you want to select.
3. Under **Products Provided**, you require **Red Hat JBoss Core Services**.

### Additional resources

- For more information about registering Red Hat Enterprise Linux 7, see the RHEL 7 Installation Guide: Subscription Manager.
- For more information about registering Red Hat Enterprise Linux 8, see the RHEL 8 Performing a Standard RHEL Installation: Registering your system using the Subscription Manager User Interface.

## 3.2. INSTALLING THE APACHE HTTP SERVER FROM RPM PACKAGES BY USING YUM

You can use the YUM package manager to install the Apache HTTP Server from RPM packages on Red Hat Enterprise Linux.

### Prerequisites

- You have attached subscriptions to Red Hat Enterprise Linux .

### Procedure

1. To subscribe to the Apache HTTP Server CDN repositories for your operating system version, enter the following command as the root user:

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

NOTE

If you are using Red Hat Enterprise Linux 7, replace ***<repository>*** with **jb-coreservices-1-for-rhel-7-server-rpms**.

If you are using Red Hat Enterprise Linux 8, replace ***<repository>*** with **jb-coreservices-1-for-rhel-8-x86_64-rpms**.

2. To install the Apache HTTP Server, enter the following command as the root user:

```
# yum groupinstall jbcs-httpd24
```

NOTE

With the release of Red Hat Enterprise Linux 8, JBCS no longer uses the yum **groupinstall** command. For more information about configuring HTTPD on Red Hat Enterprise Linux 8, see Deploying different types of servers: Setting up the Apache HTTP Web Server.

## 3.3. USING MOD_JK AND MOD_CLUSTER WITH RHEL 8

You can use the YUM package manager to install the JBoss Core Services (JBCS) **mod_jk**, **mod_cluster**, **mod_rt**, and **mod_bmx** modules on Red Hat Enterprise Linux 8.

### Procedure

1. To install **httpd**, enter the following command as the root user:

```
$ yum install httpd
```

2. To install the **mod_jk**, **mod_cluster**, **mod_rt**, and **mod_bmx** modules, enter the following commands as the root user:

```
$ yum install jbcs-httpd24-mod_jk-ap24
$ yum install jbcs-httpd24-mod_cluster-native

$ yum install jbcs-httpd24-mod_bmx

$ yum install jbcs-httpd24-mod_rt
```

> NOTE
>
> When the Apache HTTP Server (**httpd**) is installed on Red Hat Enterprise Linux 8, the base operating system modules are located in the **/usr/lib64/httpd/modules** directory. JBCS modules such as **mod_jk**, **mod_cluster**, **mod_rt**, and **mod_bmx** are currently located in the **/opt/rh/jbcs/root/usr/lib64/httpd/modules** directory. These JBCS modules follow all JBCS rules for naming, directories, and prefixes.

3. To use these modules, perform either of the following steps:

    - Create or modify the configuration file to add the **LoadModule** command. For example:

    ```
    LoadModule jk_module /opt/rh/jbcs/root/usr/lib64/httpd/modules/mod_jk.so
    ```

    - Include the directory of the installed JBCS modules in the ***JBCS_HOME*/httpd/conf.d** directory.

> NOTE
>
> - You must disable **mod_proxy_balancer** when using **mod_proxy_cluster**.
> - You must enable **mod_proxy** when using **mod_proxy_cluster**.
> - If you want **mod_proxy_cluster** to use AJP, you must disable **proxy_ajp_module**.

## 3.4. CONFIGURING THE APACHE HTTP SERVER INSTALLATION WHEN INSTALLED FROM RPMS

When you install the Apache HTTP Server from an RPM package, you can optionally remove SSL support before you run the Apache HTTP Server. The Apache HTTP Server supports SSL by default. You can choose to remove SSL support by removing the **mod_ssl** package.

### Procedure

- On a command line, enter the following command as the root user:

```
# yum remove jbcs-httpd24-mod_ssl
```

> NOTE
>
> With the release of Red Hat Enterprise Linux 8, JBCS no longer uses the yum **groupinstall** command. For more information about configuring HTTPD on Red Hat Enterprise Linux 8, see Deploying different types of servers: Setting up the Apache HTTP Web Server.

## 3.5. STARTING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM RPMS

When you install JBoss Core Services Apache HTTP Server from RPM packages, you can use the command line to start the Apache HTTP Server.

### Procedure

- On a command line, start the Apache HTTP Server service as the root user:
    - For Red Hat Enterprise Linux 7:

      ```
      # systemctl start jbcs-httpd24-httpd.service
      ```

      NOTE

      With the release of Red Hat Enterprise Linux 8, JBCS no longer uses the yum **groupinstall** command. For more information about configuring HTTPD on Red Hat Enterprise Linux 8, see Deploying different types of servers: Setting up the Apache HTTP Web Server.

IMPORTANT

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

## 3.6. STOPPING THE APACHE HTTP SERVER FROM THE COMMAND LINE WHEN INSTALLED FROM RPMS

When you install JBoss Core Services Apache HTTP Server from RPM packages, you can use the command line to stop the Apache HTTP Server.

### Procedure

- On a command line, stop the Apache HTTP Server service as the root user:
    - For Red Hat Enterprise Linux 7:

      ```
      # systemctl stop jbcs-httpd24-httpd.service
      ```

      NOTE

      With the release of Red Hat Enterprise Linux 8, JBCS no longer uses the yum **groupinstall** command. For more information about configuring HTTPD on Red Hat Enterprise Linux 8, see Deploying different types of servers: Setting up the Apache HTTP Web Server.

IMPORTANT

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

## 3.7. CONFIGURING THE APACHE HTTP SERVER SERVICE TO START AT SYSTEM STARTUP

When you install JBoss Core Services Apache HTTP Server from RPM packages, you can configure the Apache HTTP Server service to start at system startup.

**Procedure**

- To enable the Apache HTTP Server service to start at system at system startup, enter the following command:
  - For Red Hat Enterprise Linux 7:

    ```
    # systemctl enable jbcs-httpd24-httpd.service
    ```

**IMPORTANT**

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

## 3.8. SELINUX POLICIES FOR THE APACHE HTTP SERVER

You can use Security-Enhanced Linux (SELinux) policies to define access controls for the Apache HTTP Server. These policies are a set of rules that determine access rights to the product.

### 3.8.1. SELinux policy information

The SELinux security model is enforced by the kernel and ensures that applications have limited access to resources such as file system locations and ports. SELinux policies ensure that any errant processes that are compromised or poorly configured are restricted or prevented from running.

The **jbcs-httpd24-httpd-selinux** packages in your Apache HTTP Server installation provide a **mod_cluster** policy. The following table contains information about the supplied SELinux policy.

**Table 3.1. 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. |

**Additional resources**

- For more information about using SELinux on Red Hat Enterprise Linux 7, see the RHEL 7 SELinux User's and Administrator's Guide.
- For more information about using SELinux on Red Hat Enterprise Linux 8, see the RHEL8 Using SELinux guide.

### 3.8.2. Enabling SELinux policies for an Apache HTTP Server RPM installation

When you install the JBoss Core Services Apache HTTP Server from RPM packages, the **jbcs-httpd2.4-httpd-selinux** package provides SELinux policies for the Apache HTTP Server. The **jbcs-httpd2.4-httpd-selinux** package is available in the **jb-coreservices-1-for-rhel-7-server-rpms** and **jb-coreservices-1-for-rhel-6-server-rpms** Content Delivery Network (CDN) repositories.

### Procedure

- Install the **jbcs-httpd2.4-httpd-selinux** package for the Red Hat Enterprise Linux version that you are using.

# CHAPTER 4. INSTALLING THE JBOSS CORE SERVICES APACHE HTTP SERVER ON MICROSOFT WINDOWS

You can install the JBoss Core Services Apache HTTP Server on Microsoft Windows from a set of archive files that you can download from the Software Downloads page on the Red Hat Customer portal.

## 4.1. DOWNLOADING AND EXTRACTING THE APACHE HTTP SERVER ARCHIVE FILE ON MICROSOFT WINDOWS

You can download the Apache HTTP Server archive files from the Software Downloads page on the Red Hat Customer Portal. You must download the archive file for the original JBoss Core Services Apache HTTP Server 2.4.37 release from the **Releases** tab on the Software Downloads page. You can also download the latest service pack release, if any, from the **Security Advisories** tab on the Software Downloads page.

> NOTE
>
> You can install the archive file with non-root privileges, provided that you have write access to the intended installation folder.

**Prerequisites**

1. Open a browser and log in to the Software Downloads page on the Red Hat Customer Portal.
2. From the **Product** drop-down menu, select **Apache HTTP Server**.
3. From the **Version** drop-down menu, select the correct JBoss Core Services version.
4. On the **Releases** tab, click **Download** next to the Red Hat JBoss Core Services Apache HTTP Server archive file that matches the platform and architecture for your system.
5. Extract the downloaded archive file to your installation directory.

   > NOTE
   >
   > On Microsoft Windows systems, install the Apache HTTP Server in the **C:\Program Files** directory.

   The **jbcs-httpd24-2.4** directory is created when you extract the archive. This directory is the top-level directory for the Apache HTTP Server. This document refers to the **jbcs-httpd24-2.4** directory as ***HTTPD_HOME***.
6. To install the latest service pack release, perform the following steps:
   a. On the Software Downloads page, click the **Security Advisories** tab.
   b. On the **Security Advisories** tab, click **Download** next to the latest Red Hat JBoss Core Services Apache HTTP Server Patch archive file that matches the platform and architecture for your system.
      For example, if you want to install the Service Pack 10 release of the Apache HTTP Server 2.4.37 on Microsoft Windows, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.37 Patch 10 for Windows Server x86_64** file.

NOTE

Service pack releases are cumulative. By downloading the latest service pack release, you also install any previous service pack releases automatically.

## 4.2. APACHE HTTP SERVER CONFIGURATION ON MICROSOFT WINDOWS

When you install JBoss Core Services Apache HTTP Server on Microsoft Windows, you can manage the Apache HTTP Server from a command prompt or by using the Computer Management tool. Before you can run the Apache HTTP Server on Microsoft Windows, you must perform the following series of configuration tasks:

- Run the Apache HTTP Server post-installation script
- Install the Apache HTTP Server service
- Configure folder permissions for the Apache HTTP Server service
- Disable or enable SSL support

### 4.2.1. Running the Apache HTTP Server post-installation script on Microsoft Windows

Before you run the Apache HTTP Server for the first time on Microsoft Windows, you must run the Apache HTTP Server post-installation script.

**Procedure**

1. Open the **Command Prompt** as an administrative user.
2. Go to the ***HTTPD_HOME*\etc** directory.
3. Enter the following command:

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

### 4.2.2. Installing the Apache HTTP Server service

Before you run the Apache HTTP Server for the first time on Microsoft Windows, you must install the Apache HTTP Server as a Windows service.

NOTE

By default, the Apache HTTP Server is configured to use port 80. If you have Microsoft Internet Information Services (IIS) installed, you must disable or reconfigure Microsoft IIS 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 Microsoft IIS ports.

**Prerequisites**

- You have run the Apache HTTP Server post-installation script.

**Procedure**

1. Open the **Command Prompt** as an administrative user.
2. Go to the ***HTTPD_HOME*\bin** directory.
3. To install the Apache HTTP Server service, enter the following command:

```
httpd -k install
```

> NOTE
>
> A firewall security dialog might be displayed to request networking access for the Apache HTTP Server. Click **Allow** to access this service from the network.

### 4.2.3. Configuring folder permissions for the Apache HTTP Server service

Before you run the Apache HTTP Server for the first time on Microsoft Windows, you must ensure that the account used to run the service has full control over the ***HTTPD_HOME*** folder and all of its subfolders.

**Prerequisites**

- You have installed the Apache HTTP Server service.

**Procedure**

1. Right-click the ***HTTPD_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.

### 4.2.4. Disabling or enabling SSL support

Before you run the Apache HTTP Server, you can choose to disable or enable SSL support by renaming the SSL configuration file. The Apache HTTP Server supports SSL by default.

### Prerequisites

- You have configured folder permissions for the Apache HTTP Server service .

### Procedure

1. Go to the ***HTTPD_HOME*\conf.d** directory.
2. To enable or disable SSL, perform either of the following steps:
   - If you want to disable SSL, rename **ssl.conf** to **ssl.conf.disabled**.
   - If you want to re-enable SSL, rename **ssl.conf.disabled** to **ssl.conf**.

## 4.3. STARTING THE APACHE HTTP SERVER ON MICROSOFT WINDOWS

When you install JBoss Core Services Apache HTTP Server on Microsoft Windows, you can start the Apache HTTP Server service by using the Command Prompt or the Computer Management tool.

### Prerequisites

- You have configured the Apache HTTP Server.

### Procedure

- Perform either of the following steps:
  - Open the Command Prompt as an administrator and enter the following command:

    ```
    net start Apache2.4
    ```

  - Click **Start** > **Administrative Tools** > **Services**, right-click the **httpd** service, and click **Start**.

## 4.4. STOPPING THE APACHE HTTP SERVER ON MICROSOFT WINDOWS

When you install JBoss Core Services Apache HTTP Server on Microsoft Windows, you can stop the Apache HTTP Server service by using the Command Prompt or the Computer Management tool.

### Prerequisites

- You have started the Apache HTTP Server.

### Procedure

- Perform either of the following steps:
  - Open the Command Prompt as an administrator and enter the following command:

```
net stop Apache2.4
```

- Click **Start** > **Administrative Tools** > **Services**, right-click the **httpd** service, and click **Stop**.

# CHAPTER 5. ENABLING HTTP/2 FOR THE JBOSS CORE SERVICES APACHE HTTP SERVER

The Hypertext Transfer Protocols (HTTP) are standard methods of transmitting data between applications, such as servers and browsers, over the internet. The Apache HTTP Server supports the use of HTTP/2 for encrypted connections that are using Transport Layer Security (TLS), which is indicated by the **h2** keyword when enabled.

HTTP/2 improves on HTTP/1.1 by providing the following enhancements:

- Header compression omits implied information to reduce the size of the header that is transmitted.
- Multiple requests and responses over a single connection use binary framing rather than textual framing to break down response messages.

NOTE

The Apache HTTP Server does not support the use of HTTP/2 for unencrypted connections that are using the Transmission Control Protocol (TCP), which is indicated by the **h2c** keyword when enabled.

HTTP/2 is not available for web servers that are using the Multi-Processing Module (MPM) pre-fork (**modules/mod_mpm_prefork.so**).

## 5.1. PREREQUISITES

- You have root user access on Red Hat Enterprise Linux.
- You have administrative access on Windows Server.
- You have installed Red Hat JBoss Core Services Apache HTTP Server 2.4.23 or later.
- You have installed the SSL module (**modules/mod_ssl.so**).
  If you need to install the SSL module, enter the following command:

  ```
  yum install mod_ssl
  ```

- You have installed the HTTP/2 module (**modules/mod_http2.so**).
  If you need to install the HTTP/2 module, enter the following command:

  ```
  yum install mod_http2
  ```

NOTE

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

## 5.2. ENABLING HTTP/2 FOR THE APACHE HTTP SERVER

You can enable HTTP/2 for the Apache HTTP Server by updating configuration file settings in the ***HTTP_HOME*** directory.

### Procedure

1. To add the **http2_module** to the configuration:

    a. Open the ***HTTP_HOME*/conf.modules.d/00-base.conf** file.

    b. Enter the following line:

    ```
    ...
    LoadModule http2_module modules/mod_http2.so
    ```

2. To add the **h2** protocol to the configuration:

    a. Open the ***HTTP_HOME*/conf/httpd.conf** file.

    b. If you want to enable HTTP/2 support for a virtual host, add the **h2** protocol to the virtual host configuration.
    Alternatively, if you want to enable HTTP/2 support for all server connections, add the **h2** protocol to the main server configuration section.

    For example:

    ```
    <IfModule http2_module>
       Protocols h2 http/1.1
       ProtocolsHonorOrder on
    </IfModule>
    ```

3. To update the Secure Socket Layer (SSL) configuration:

    a. Open the ***HTTP_HOME*/conf.d/ssl.conf** file:

    b. Ensure the **SSLEngine** directive is set to enabled. The SSL Engine is enabled by default.

    ```
    SSLEngine on
    ```

    c. Update the **SSLProtocol** directive to disable the **SSLv2** and **SSLv3** protocols. This forces connections to use the Transport Layer Security (TLS) Protocols.

    ```
    SSLProtocol all -SSLv2 -SSLv3
    ```

    d. Update the **SSLCipherSuite** directive to specify which SSL ciphers can be used with the Apache HTTP Server.
    For example:

    ```
    SSLCipherSuite ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-DSS-AES128-GCM-SHA256:kEDH+AESGCM:ECDHE-RSA-AES128-SHA256:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-RSA-AES256-SHA384:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA:DHE-DSS-AES128-SHA256:DHE-RSA-AES256-SHA256:DHE-DSS-AES256-SHA:DHE-RSA-AES256-SHA:!aNULL:!eNULL:!EXPORT:!DES:!RC4:!3DES:!MD5:!PSK
    ```

> **NOTE**
>
> For more information about the SSL module and the supported directives, see Apache HTTP Server Documentation Version 2.4 - Modules: Apache Module mod_ssl.

4. To restart the Red Hat JBoss Core Services Apache HTTP Server, and apply the changed configuration, perform one of the following steps as the root user:

   - If you want to use **systemd** to start the Apache HTTP Server on Red Hat Enterprise Linux, enter the following command:

     ```
     # systemctl restart jbcs-httpd24-httpd.service
     ```

   - If you want to use **apachectl** to start Red Hat JBoss Core Services on Red Hat Enterprise Linux, enter the following command:

     ```
     # HTTP_HOME/sbin/apachectl restart
     ```

   - If you want to start the Apache HTTP Server on Windows Server, enter the following command:

     ```
     # net restart Apache2.4
     ```

### Additional resources

- For more information about the HTTP/2 module and the supported directives, see Apache HTTP Server Documentation Version 2.4 - Modules: Apache Module mod_http2.
- For more information about the SSL module and the supported directives, see Apache HTTP Server Documentation Version 2.4 - Modules: Apache Module mod_ssl.

## 5.3. VIEWING APACHE HTTP SERVER LOGS TO VERIFY THAT HTTP/2 IS ENABLED

You can view the Apache HTTP Server access log or request log to verify that HTTP/2 is enabled.

### Prerequisites

- You have enabled HTTP/2.

### Procedure

1. Access the server from a browser or by using the **curl** command-line tool.
2. To check the SSL/TLS request log, enter the following command:

   ```
   $ grep 'HTTP/2'  HTTP_HOME/logs/ssl_request_log
   ```

3. To check the SSL/TLS access log, enter the following command:

   ```
   $ grep 'HTTP/2'  HTTP_HOME/logs/ssl_access_log
   ```

### Verification

1. If HTTP/2 is enabled, the **grep 'HTTP/2' *HTTP_HOME*/logs/ssl_request_log** command produces the following type of output:

   ```
   [26/Apr/2018:06:44:45 +0000] 172.17.0.1 TLSv1.2 AES128-SHA "HEAD /html-single/index.html HTTP/2" -
   ```

2. If HTTP/2 is enabled, the **grep 'HTTP/2' *HTTP_HOME*/logs/ssl_access_log** command produces the following type of output:

   ```
   172.17.0.1 - - [26/Apr/2018:06:44:45 +0000] "HEAD /html-single/index.html HTTP/2" 200 -
   ```

## 5.4. USING THE CURL COMMAND TO VERIFY THAT HTTP/2 IS ENABLED

You can use the **curl** command-line tool to verify that HTTP/2 is enabled.

**NOTE**

The **curl** package that is provided with Red Hat Enterprise Linux 7 or earlier does not support HTTP/2.

### Prerequisites

- You have enabled HTTP/2.
- You are using a version of **curl** that supports **HTTP2**.
  To check that you are using a version of **curl** that supports HTTP/2, enter the following command:

  ```
  $ curl -V
  ```

  This command produces the following type of output:

  ```
  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
  ```

### Procedure

1. To check that the HTTP/2 protocol is active, enter the following command:

   ```
   $ curl -I http://<JBCS_httpd_server>:<port>/<test.html>
   ```

NOTE

In the preceding example, replace ***<JBCS_httpd_server>*** with the URI of the server, such as **example.com**, and replace ***<test.html>*** with any HTML file that you want to use to test the configuration. An example HTML test page is not provided. The port number is dependent on your configuration.

Verification

- If the HTTP/2 protocol is active, the **curl** command produces the following output:

```
HTTP/2 200
```

  Otherwise, if the HTTP/2 protocol is inactive, the **curl** command produces the following output:

```
HTTP/1.1 200
```

## 5.5. ADDITIONAL RESOURCES (OR NEXT STEPS)

- For more information about using HTTP/2, see Apache HTTP Server Documentation Version 2.4 - How-To / Tutorials: HTTP/2 guide.
- For information about SSL configuration, see Apache HTTP Server Documentation Version 2.4 - SSL/TLS Strong Encryption: How-To.
- For more information about the proposed internet standard for HTTP/2, see IETF: RFC 7540 - Hypertext Transfer Protocol Version 2 (HTTP/2).

*Revised on 2023-01-04 15:04:40 UTC*