Red Hat JBoss Core Services 2.4.51

Apache HTTP Server Installation Guide

For use with Red Hat JBoss middleware products.

Last Updated: 2024-02-06

Red Hat JBoss Core Services 2.4.51 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

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

Procedure

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

MAKING OPEN SOURCE MORE INCLUSIVE

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

CHAPTER 1. INTRODUCTION TO JBCS APACHE HTTP SERVER INSTALLATION

Red Hat JBoss Core Services (JBCS) provides a collection of supplementary software, including the Apache HTTP Server, that you can use with various Red Hat JBoss middleware products. Red Hat packages this supplementary software under JBCS to allow for faster distribution of updates and for a more consistent update experience.

For a full list of components that JBCS supports, 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, ensure that you have an active Red Hat subscription and you are logged in to the Red Hat Customer Portal.

1.1. JBCS APACHE HTTP SERVER

Red Hat JBoss Core Services (JBCS) provides a distribution of the Apache HTTP Server that multiple Red Hat JBoss middleware products use. The Apache HTTP Server processes requests that web clients send over the Hypertext Transfer Protocol (HTTP).

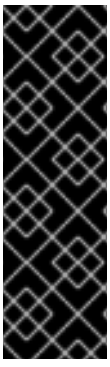
Apache HTTP Server distributions for JBoss middleware products

In older JBoss product releases, each JBoss middleware product provided a separate distribution of the Apache HTTP Server. Starting from the following product versions, each JBoss middleware product uses the JBCS distribution of the Apache HTTP Server:

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

Differences between JBCS and RHEL distributions of the Apache HTTP Server

Both JBCS and Red Hat Enterprise Linux (RHEL) provide separate distributions of the Apache HTTP Server.



IMPORTANT

On RHEL 9, JBCS does not provide an RPM distribution of the Apache HTTP Server. JBCS provides only an archive file distribution of the Apache HTTP Server for RHEL 9 systems.

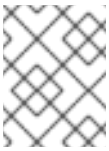
Unlike JBCS releases on earlier RHEL versions, the JBCS distribution of the Apache HTTP Server for RHEL 9 systems is based on the RHEL distribution of the Apache HTTP Server **httpd** package. JBCS provides an archive file distribution on RHEL 9 to support the ability to run multiple instances of the Apache HTTP Server simultaneously.

Consider the following differences between the Apache HTTP Server distributions that JBCS and RHEL provide:

On RHEL versions 7 and 8

- You can install the JBCS Apache HTTP Server from an archive file or RPM package. You can install the RHEL Apache HTTP Server from an RPM package only.

- Only the JBCS Apache HTTP Server provides the load-balancing HTTP connectors **mod_jk** and **mod_proxy_cluster**. The RHEL Apache HTTP Server does not provide these modules.



NOTE

Prior to the JBCS 2.4.51 release, the **mod_proxy_cluster** connector was named **mod_cluster**.

- On RHEL 7, only the JBCS Apache HTTP Server provides the **mod_proxy_uwsgi** module. From RHEL 8 onward, the JBCS and RHEL distributions of the Apache HTTP Server both provide the **mod_proxy_uwsgi** module.

On RHEL 9

- Unlike JBCS releases on RHEL 7 and RHEL 8, the JBCS release on RHEL 9 is based on the RHEL distribution of the Apache HTTP Server **httpd** package. JBCS on RHEL 9 therefore has certain behavioral differences compared to the JBCS distributions of the Apache HTTP Server on earlier RHEL versions. For more information, see Behavioral differences between JBCS distributions on different RHEL versions.

- JBCS provides only an archive file distribution of the Apache HTTP Server. If you want to install the Apache HTTP Server from an RPM package, your only option is to install the RHEL distribution of the **httpd** package by using Application Streams.

- The version of the Apache HTTP Server that JBCS provides is different from the version of the Apache HTTP Server that RHEL provides through the Application Streams feature.

- The JBCS and RHEL distributions of the Apache HTTP Server provide identical copies of the **mod_jk** connector and the **mod_proxy_cluster** connector.

On all RHEL versions

- The JBCS Apache HTTP Server uses a top-level **jbcs-httpd24-2.4/httpd** installation directory. The RHEL Apache HTTP Server uses standard RHEL directories for an installation of the **httpd** package such as **/etc/httpd**, **usr/share/httpd**, **var/log/httpd**, and so on.

- When you install a JBCS distribution of the Apache HTTP Server from an archive file or from RPM packages by using the **groupinstall** option, you also automatically install the **mod_jk** and **mod_proxy_cluster** connectors.

- The JBCS Apache HTTP Server does not provide or support the **mod_php** module. Only the RHEL Apache HTTP Server supports the **mod_php** module.

Behavioral differences between JBCS distributions on different RHEL versions

Unlike JBCS 2.4.51 on RHEL 7 or RHEL 8, the JBCS 2.4.51 distribution for RHEL 9 systems is based on the RHEL distribution of the Apache HTTP Server **httpd** package. This change to the way Red Hat distributes the **httpd** package from RHEL 9 onward helps to provide Apache HTTP Server users with a more consistent and streamlined user experience.

Because of this difference, JBCS 2.4.51 on RHEL 9 has certain behavioral differences compared to JBCS 2.4.51 on earlier RHEL versions.

Consider the following guidelines:

- On RHEL 9, the **mod_security** module does not support the **SecCollectionGCFrequency** directive for specifying garbage collection frequency. The **mod_security** module that JBCS provides on RHEL 7 and RHEL 8 supports the **SecCollectionGCFrequency** directive.
- On RHEL 9, the **mod_deflate** module does not support the **DeflateAlterEtag** directive for specifying how to alter the ETag header when a response is compressed. The **mod_deflate** module that JBCS provides on RHEL 7 and RHEL 8 supports the **DeflateAlterEtag** directive.
- On RHEL 9, the **httpd.conf.sample** file does not include the following content:
  - A default **PidFile** directive for specifying the file in which the server records the process ID of the daemon
  - A list of **AddLanguage** directives in the **mod_mime** section for mapping specific filename extensions to specific content languages
  - A configuration section for the **web_dav** module for web-based distributed authoring and versioning (WebDav)

  The **httpd.conf.sample** file that JBCS provides on RHEL 7 and RHEL 8 includes all of the preceding content.

1.2. SUPPORTED OPERATING SYSTEMS AND INSTALLATION METHODS FOR THE JBCS APACHE HTTP SERVER

Red Hat JBoss Core Services (JBCS) provides a distribution of the Apache HTTP Server for different versions of the Red Hat Enterprise Linux (RHEL) and Windows Server operating systems.

Consider the following guidelines for installing the JBCS Apache HTTP Server on supported operating systems:

- On all supported RHEL and Windows Server versions, you can install the JBCS Apache HTTP Server by using archive installation files that are available for each platform.
- On RHEL versions 7 and 8, you can install the JBCS Apache HTTP Server by using Red Hat Package Manager (RPM) packages.
- On RHEL 9, you *cannot* install the JBCS Apache HTTP Server by using RPM packages. If you want to install the Apache HTTP Server from an RPM package on RHEL 9, your only option is to install the RHEL distribution of the Apache HTTP Server by using Application Streams.

Additional resources

- Core Services HTTP Server Supported Configurations web page

1.3. UPGRADE OF AN EXISTING JBCS INSTALLATION TO THE 2.4.51 RELEASE

If you previously installed Red Hat JBoss Core Services (JBCS) 2.4.37 or earlier, you can upgrade your existing JBCS installation to the latest 2.4.51 release. The steps to upgrade JBCS differ depending on whether you installed the product from archive files or RPM packages.

1.3.1. Upgrading an existing JBCS installation when installed from archive files

If you previously installed the JBCS Apache HTTP Server 2.4.37 or earlier from an archive file, you can upgrade to the latest 2.4.51 release.

The upgrade process includes the following steps:

1. Installing the Apache HTTP Server 2.4.51
2. Setting up the Apache HTTP Server 2.4.51
3. Removing an earlier version of Apache HTTP Server

Prerequisites

- If you are using Red Hat Enterprise Linux (RHEL), you have root user access.
- If you are using Windows Server, you have administrative access.
- You have an existing installation of the JBCS Apache HTTP Server 2.4.37 or earlier that you installed from an archive file.

Procedure

1. Shut down any running instances of the Apache HTTP Server 2.4.37.
2. Back up the Apache HTTP Server 2.4.37 installation and configuration files.
3. Install the Apache HTTP Server 2.4.51 by using the archive file 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 2.4.37 to version 2.4.51.

NOTE

The JBCS configuration files might have changed since the Apache HTTP Server 2.4.37 release. Update the 2.4.51 version configuration files rather than overwrite them with the configuration files from a different version, such as the Apache HTTP Server 2.4.37.

5. Remove the Apache HTTP Server 2.4.37 root directory.

Additional resources

- Installing the JBCS Apache HTTP Server on RHEL from archive files
- Installing the JBCS Apache HTTP Server on Windows Server

1.3.2. Upgrading an existing JBCS installation when installed from RPM packages

If you previously installed the JBCS Apache HTTP Server 2.4.37 or earlier from RPM packages, you can upgrade to the latest 2.4.51 release by using the **yum groupupdate** command.

Prerequisites

- You have an existing installation of the JBCS Apache HTTP Server 2.4.37 or earlier that you installed from RPM packages on RHEL 7 or RHEL 8.

Procedure

- Enter the following command as the root user:

```
# yum groupupdate jbcs-httpd24
```

Additional resources

- Installing the JBCS Apache HTTP Server on RHEL from RPM packages

1.4. KEY DIFFERENCES BETWEEN RHEL 7 AND RHEL 8

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

Removed security functionality

All-numeric user and group names are deprecated in RHEL 7 and their support is completely removed in RHEL 8.

Memory management

In RHEL 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 RHEL 8 supports the implementation of 5-level page tables, to allow handling of the expanded address range. In RHEL 8, support for 5-level page tables is disabled by default, even if the system supports this feature.

XFS supports

RHEL 7 can mount XFS file systems with shared copy-on-write data extents only in the read-only mode. In RHEL 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 RHEL 7, the NFS configuration is located in the **/etc/sysconfig/nfs** file. In RHEL 8, the NFS configuration is located in the **/etc/nfs.conf** file.

Additional resources

- Considerations in adopting RHEL 8

1.5. KEY DIFFERENCES BETWEEN RHEL 8 AND RHEL 9

This section provides an overview of some of the key changes introduced in Red Hat Enterprise Linux (RHEL) 9.

Application Streams enhancement

RHEL 8 introduced a feature called *Application Streams*. RHEL uses Application Streams to deliver and update multiple versions of user-space components such as applications, runtime languages, and databases more frequently than the core operating system packages. Each Application Stream represents a specific version of a component, and each component in an Application Stream has a defined life cycle. Application Streams provide users greater flexibility to use the component versions that suit their requirements for specific use cases and workloads without impacting the underlying stability of the platform or deployments.

On RHEL 8, Red Hat packaged the content in Application Streams as a combination of RPM

packages, modules (package groups), and Software Collections. RHEL 9 further enhances the Application Streams feature by providing initial Application Stream versions that you can install as RPM packages by using the standard **dnf install** command

Availability of Apache connectors and load balancers

RHEL 9 provides a distribution of the Apache Tomcat Connector (**mod_jk**) and the JBoss HTTP Connector (**mod_proxy_cluster**) for load-balancing web client requests to back-end application servers. The RHEL distribution of **mod_jk** and **mod_proxy_cluster** is identical to the JBCS distribution of these modules.

Installing the RHEL distribution of the Apache HTTP Server does not automatically install the **mod_jk** and **mod_proxy_cluster** modules. For more information about installing **mod_jk** and **mod_proxy_cluster** from RPM packages on RHEL 9, see the Apache HTTP Server Connectors and Load Balancing Guide.

Additional resources

- Considerations in adopting RHEL 9

1.6. ADDITIONAL RESOURCES (OR NEXT STEPS)

- Installing the JBCS Apache HTTP Server on RHEL from archive files
- Installing the JBCS Apache HTTP Server on RHEL 7 or RHEL 8 from RPM packages
- Installing the JBCS Apache HTTP Server on Windows Server
- Installing the Apache HTTP Server on RHEL 9 by using Application Streams

CHAPTER 2. INSTALLING THE JBCS APACHE HTTP SERVER ON RHEL FROM ARCHIVE FILES

On Red Hat Enterprise Linux (RHEL) versions 7, 8, and 9, Red Hat JBoss Core Services (JBCS) provides a distribution of the Apache HTTP Server that you can install from archive files. You can download and extract the archive files from the Software Downloads page on the Red Hat Customer Portal. You must install the base archive file for the original 2.4.51 release. You can also install the latest service pack release, if any.

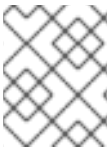
When you install the Apache HTTP Server from an archive file, you can manage the product in different ways. For example, you can use a system daemon at system startup or manage the Apache HTTP Server from a command line.

NOTE

From the 2.4.51 Service Pack 2 release onward, JBCS supports installation of the Apache HTTP Server from archive files on RHEL 9. For JBCS Apache HTTP Server installations on RHEL 9, the supported Apache HTTP Server version is 2.4.53.

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 files are slightly different.



NOTE

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

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 JBCS version.
4. Depending on the RHEL version that you are using, perform one of the following steps:
   - If you are using RHEL 7, on the **Releases** tab, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 for RHEL 7 x86_64** file.
   - If you are using RHEL 8, on the **Releases** tab, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 for RHEL 8 x86_64** file.

- If you are using RHEL 9, click the **Security Advisories** tab. Then click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 Patch 02 for RHEL 9 x86_64** file.

NOTE

The **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 Patch 02 for RHEL 9 x86_64** file is the base archive file for installing the JBCS Apache HTTP Server on RHEL 9.

Despite the **2.4.51** naming convention, the JBCS archive file for RHEL 9 provides a distribution of Apache HTTP Server 2.4.53.

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

NOTE

On RHEL systems, install the Apache HTTP Server in the **/opt/** directory.

The extraction of the archive file automatically creates the top-level **jbcs-httpd24-2.4/httpd** 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 2.4.51 Patch** archive file that matches the platform and architecture for your system.
   For example, if you want to install the Service Pack *X* release of the Apache HTTP Server 2.4.51 on RHEL 8, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 Patch *X* for RHEL 8 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 JBCS Apache HTTP Server from an archive file on RHEL, 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 account and 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

- 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 the JBCS Apache HTTP Server from an archive file on RHEL, 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 the JBCS Apache HTTP Server from an archive file on RHEL, 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 PRIVILEGES

When you install the JBCS Apache HTTP Server from an archive file on RHEL, you can start the Apache HTTP Server from the command line as a user without root privileges. In this situation, you can use a non-root user account, such as the **apache** user.

Procedure

1. Stop all instances of the Apache HTTP Server :

```
pkill httpd
```

2. In the ***HTTPD_HOME*/conf/httpd.conf** file, set the **http** listen port to higher than 1024:

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

3. In the ***HTTPD_HOME*/conf.d/ssl.conf** file, set the **https** listen port to higher than 1024:

```
Listen 2443
```

4. Change the ownership of the **logs** directory:

```
chown -R apache:apache HTTPD_HOME/logs/
```

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

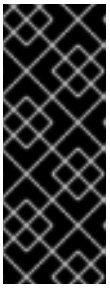
```
chown -R apache:apache HTTPD_HOME/var/run/
```

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 and enable SELinux . 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 JBCS Apache HTTP Server from an archive file on RHEL, you can use a system daemon to perform management tasks. Using the Apache HTTP Server with a system daemon provides a way to start the Apache HTTP Server services at system startup. The system daemon also provides start, stop and status check functions.

On RHEL versions 7, 8, and 9, the default system daemon is **systemd**.

IMPORTANT

RHEL 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 by using **systemd**, enter any of the following commands as the root user:

   - To enable the Apache HTTP Server services to start at system startup:

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

   - To start the Apache HTTP Server:

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

   - To stop the Apache HTTP Server:

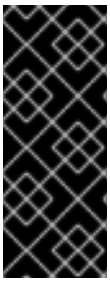
```
# systemctl stop jbcs-httpd24-httpd.service
```

   - To verify the status of the Apache HTTP Server:

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

NOTE

Any user can run the **systemctl status** command.



IMPORTANT

To revert any changes that the **.postinstall.systemd** script affects, you can enter the following command:

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

For more information about using **systemd**, see the **Additional resources** links.

Additional resources

- RHEL 7: System Administrator's Guide: Managing System Services
- RHEL 8: Configuring Basic System Settings: Managing system services with systemctl
- RHEL 9: 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_proxy_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_proxy_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_proxy_cluster** to enable the **httpd** process to write at this location.

Additional resources

- RHEL 7: SELinux User's and Administrator's Guide
- RHEL 8: Using SELinux
- RHEL 9: Using SELinux

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

In this release, the archive packages provide SELinux policies. The root Apache HTTP Server folder includes a **.postinstall.selinux** file. 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 JBCS APACHE HTTP SERVER ON RHEL 7 OR RHEL 8 FROM RPM PACKAGES

On Red Hat Enterprise Linux (RHEL) versions 7 and 8, Red Hat JBoss Core Services (JBCS) provides a distribution of the Apache HTTP Server that you can install from RPM packages. RPM installation packages for the JBCS Apache HTTP Server are available from Red Hat Subscription Management. Installing the Apache HTTP Server from RPM packages installs the Apache HTTP Server as a service.

IMPORTANT

JBCS provides RPM distributions of the Apache HTTP Server for RHEL versions 7 and 8 only. JBCS does not provide an RPM distribution of the Apache HTTP Server for RHEL 9.

If you want to install the Apache HTTP Server from RPM packages on RHEL 9, you must use the Application Streams feature of RHEL. For more information, see Installing the Apache HTTP Server on RHEL 9 by using Application Streams.

3.1. ATTACHING SUBSCRIPTIONS TO RHEL

Before you download and install the RPM packages for the Apache HTTP Server, you must attach subscriptions to Red Hat Enterprise Linux (RHEL). You can attach subscriptions by registering your system with Red Hat Subscription Management and by subscribing 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**.

For more information about registering your installed version of RHEL, see the **Additional resources** links.

Additional resources

- RHEL 7: Installation Guide: Subscription Manager.
- RHEL 8: Configuring basic system settings: Registering the system and managing subscriptions .

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

You can install the JBCS Apache HTTP Server from RPM packages on RHEL 7 or RHEL 7 by using the YUM package manager.

Prerequisites

- You have attached subscriptions to RHEL .

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 RHEL 7, replace ***<repository>*** with **jb-coreservices-1-for-rhel-7-server-rpms**.

If you are using RHEL 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
```

3.3. 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
```

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

When you install JBCS 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:

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

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

When you install JBCS 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:

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

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

When you install JBCS 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 startup, enter the following command as the root user:

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

3.7. 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.

3.7.1. 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_proxy_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_proxy_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_proxy_cluster** to enable the **httpd** process to write at this location.

Additional resources

- RHEL 7: SELinux User's and Administrator's Guide
- RHEL 8: Using SELinux

3.7.2. Enabling SELinux policies for an Apache HTTP Server RPM installation

When you install the JBCS 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-8-x86_64-rpms** Content Delivery Network (CDN) repositories.

Procedure

- Install the **jbcs-httpd2.4-httpd-selinux** package for the RHEL version that you are using.

CHAPTER 4. INSTALLING THE JBCS APACHE HTTP SERVER ON WINDOWS SERVER

You can install the JBCS Apache HTTP Server on Windows Server 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 WINDOWS SERVER

You can download the Apache HTTP Server archive files from the Software Downloads page on the Red Hat Customer portal. You can download the archive file for the base JBCS Apache HTTP Server 2.4.51 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

If you have write access to the intended installation folder, you can install the archive file with non-administrator privileges.

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 JBCS version.
4. On the **Releases** tab, click **Download** next to the JBCS 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 Windows Server systems, install the Apache HTTP Server in the **C:\Program Files** directory.

   The extraction of the archive file automatically creates the top-level **jbcs-httpd24-2.4** folder for the Apache HTTP Server. This document refers to the **jbcs-httpd24-2.4** folder 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 2.4.51 Patch** archive file that matches the platform and architecture for your system.
   For example, if you want to install the Service Pack *X* release of the Apache HTTP Server 2.4.51 on Windows Server, click **Download** next to the **Red Hat JBoss Core Services Apache HTTP Server 2.4.51 Patch *X* 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 WINDOWS SERVER

When you install JBCS Apache HTTP Server on Windows Server, 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 Windows Server, 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 Windows Server

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

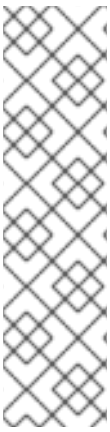
Procedure

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 Windows Server, 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 Windows Server, 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 WINDOWS SERVER

When you install JBCS Apache HTTP Server on Windows Server, 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 WINDOWS SERVER

When you install JBCS Apache HTTP Server on Windows Server, 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

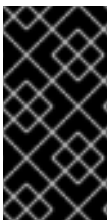
- Perform either of the following steps:
  - Open the Command Prompt as an administrator and enter the following command:

    ```
    net stop Apache2.4
    ```

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

CHAPTER 5. INSTALLING THE APACHE HTTP SERVER ON RHEL 9 BY USING APPLICATION STREAMS

The Red Hat Enterprise Linux (RHEL) Application Streams feature delivers and updates multiple versions of user-space components such as applications, runtime languages, and databases in an **AppStream** repository. On RHEL 9, you can install the RHEL distribution of the Apache HTTP Server from an RPM package by using Application Streams.



IMPORTANT

Red Hat JBoss Core Services (JBCS) does not provide an RPM distribution of the Apache HTTP Server for RHEL 9. The Apache HTTP Server **httpd** package that the RHEL **AppStream** repository provides is the only supported RPM distribution of the Apache HTTP Server for RHEL 9 systems.

NOTE

The RHEL **AppStream** repository currently provides only one version of the Apache HTTP Server. The supported **httpd** package version in the RHEL **AppStream** repository is 2.4.53 or later.

Installing the RHEL distribution of the Apache HTTP Server does not automatically install the **mod_jk** and **mod_proxy_cluster** packages. For more information about installing **mod_jk** and **mod_proxy_cluster** from RPM packages on RHEL 9, see the Apache HTTP Server Connectors and Load Balancing Guide.

5.1. INSTALLATION OF THE APACHE HTTP SERVER WHEN USING APPLICATION STREAMS

You can install the RHEL 9 distribution of the Apache HTTP Server from an RPM package by using the standard **dnf install** command. You can subsequently start and stop the Apache HTTP Server from the command line as the root user. Alternatively, you can enable the Apache HTTP Server to start automatically at system startup.

For more information about installing, starting, and stopping the RHEL distribution of the Apache HTTP Server, see Setting up the Apache HTTP web server.

Additional resources

- Application Streams
- Managing software with the DNF tool

5.2. 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.

The Apache HTTP Server has an SELinux type name of **httpd_t**. By default, the Apache HTTP Server can access files and directories in **/var/www/html** and other web server directories that have an SELinux type context of **httpd_sys_content_t**.

You can also customize the SELinux policy for the Apache HTTP Server if you want to use a non-standard configuration.

Additional resources

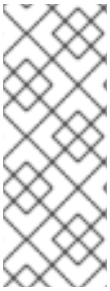
- Using SELinux
- Customizing the SElinux policy for the Apache HTTP Server in a non-standard configuration

CHAPTER 6. ENABLING HTTP/2 FOR THE JBCS 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**).

6.1. 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.

6.2. 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.

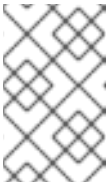
```
SSLEngine on
```

   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.

6.3. 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 -
```

6.4. 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
```

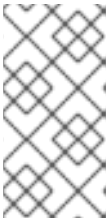
  This command produces the following type of output:

```
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 https://<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
```

6.5. 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).

CHAPTER 7. SECURING CONNECTIONS BY USING OCSP

Online Certificate Status Protocol (OCSP) is a technology that allows web browsers and web servers to communicate over a secured connection. The encrypted data is sent from one side and decrypted by the other side before processing. The web browser and the web server both encrypt and decrypt the data.

7.1. ONLINE CERTIFICATE STATUS PROTOCOL

When a web browser and a web server communicate over a secured connection, the server presents a set of credentials in the form of a certificate. The browser then validates the certificate and sends a request for certificate status information. The server responds with a certificate status of current, expired, or unknown.

The certificate contains the following types of information:

- Syntax for communication
- Control information such as start time, end time, and address information to access an Online Certificate Status Protocol (OCSP) responder.

The web server uses an OCSP responder to check the certificate status. You can configure the web server to use the OCSP responder that is listed in the certificate or another OCSP responder. OCSP allows a grace period for expired certificates, which allows access to a server for a limited time before renewing the certificate.

OCSP overcomes limitations of the older Certificate Revocation List (CRL) method.

Additional resources

- Red Hat Certificate System *Planning, Installation, and Deployment Guide*.

7.2. CONFIGURING THE APACHE HTTP SERVER FOR SSL CONNECTIONS

You can configure the Apache HTTP Server to support SSL connections, by installing the **mod_ssl** package and specifying configuration settings in the **ssl.conf** file.

Prerequisites

- You have generated an SSL certificate and private key.
- You know the location of the SSL certificate and private key file.
- You have obtained the Common Name (CN) that is associated with the SSL certificate.

Procedure

1. To install **mod_ssl**, enter the following command:

```
# yum install jbcs-httpd24-mod_ssl
```

2. To specify SSL configuration settings:

a. Open the ***JBCS_HOME*/httpd/conf.d/ssl.conf** file.

b. Enter details for the **ServerName**, **SSLCertificateFile**, and **SSLCertificateKeyFile**. For example:

```
<VirtualHost _default_:443>
ServerName www.example.com:443
SSLCertificateFile /opt/rh/jbcs-httpd24/root/etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /opt/rh/jbcs-httpd24/root/etc/pki/tls/private/localhost.key
```

NOTE

- The **ServerName** must match the Common Name (CN) that is associated with the SSL certificate. If the **ServerName** does not match the CN, client browsers display domain name mismatch errors.
- The **SSLCertificateFile** specifies the path to the SSL certificate file.
- The **SSLCertificateKeyFile** specifies the path to the private key file that is associated with the SSL certificate.

3. Verify that the **Listen** directive matches the hostname or IP address for the **httpd** service for your deployment.

4. To restart the Apache HTTP Server, enter the following command:

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

7.3. USING OCSP WITH THE APACHE HTTP SERVER

You can use the Online Certificate Status Protocol (OCSP) for secure connections with the Apache HTTP Server.

Prerequisites

- You have configured Apache HTTP Server for SSL connections.

Procedure

1. Configure a certificate authority.

NOTE

Ensure that your CA can issue OCSP certificates. The CA must be able to append the following attributes to the certificate:

```
[ usr_cert ]
...
authorityInfoAccess=OCSP;URI:http://<HOST>:<PORT>
...
[ v3_OCSP ]
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = OCSP Signing
```

In the preceding example, replace ***HOST*** and ***PORT*** with the details of the OCSP responder that you will configure.

2. Configure an OCSP responder.

Additional resources

- Managing certificates and certificate authorities.
- Configuring OCSP responders.

7.4. CONFIGURING THE APACHE HTTP SERVER TO VALIDATE OCSP CERTIFICATES

You can configure the Apache HTTP Server to validate OCSP certificates, by defining OCSP settings in the **ssl_conf** file.

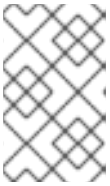
Prerequisites

- You have configured a Certificate Authority (CA) .
- You have configured an OCSP Responder.

Procedure

1. Open the ***JBCS_HOME*/httpd/conf.d/ssl.conf** file.
2. Specify the appropriate OCSP configuration details for your deployment.
   For example:

```
# Require valid client certificates (mutual auth)
SSLVerifyClient require
SSLVerifyDepth  3
# Enable OCSP
SSLOCSPEnable on
SSLOCSPDefaultResponder http://<HOST>:<PORT>
SSLOCSPOverrideResponder on
```



NOTE

The preceding example shows how to enable OCSP validation of client certificates. In the preceding example, replace ***<HOST>*** and ***<PORT>*** with the IP address and port of the default OCSP Responder.

7.5. VERIFYING THE OCSP CONFIGURATION FOR THE APACHE HTTP SERVER

You can use the OpenSSL command-line tool to verify the OCSP configuration for the Apache HTTP Server.

Procedure

- On the command line, enter the **openssl** command in the following format:

```
# openssl ocsp -issuer cacert.crt -cert client.cert -url http://HOST:PORT -CA ocsp_ca.cert -VAfile ocsp.cert
```

  In the preceding command, ensure that you specify the following details:

  - Use the **-issuer** option to specify the CA certificate.
  - Use the **-cert** option to specify the client certificate that you want to verify.
  - Use the **-url** option to specify the HTTP server validating Certificate (OCSP).
  - Use the **-CA** option to specify the CA certificate for verifying the Apache HTTP Server server certificate.
  - Use the **-VAfile** option to specify the OCSP responder certificate.

Revised on 2024-02-06 12:39:59 UTC