

# HTTP SERVERS

Adapted from Linux Foundation

# 7.1 Objective

**By the end of this session, you should be able to:**

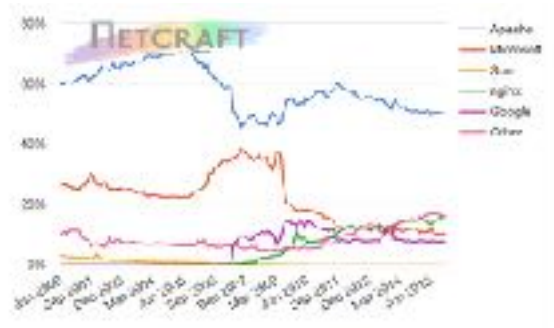
- **Install and configure an Apache web server supporting:**
- **Default configuration.**
- **IP-based virtual hosts.**
- **Name-based virtual hosts.**
- **Directory-level security.**
- **SSL encryption for a host.**

## 7.2 Apache History

**Apache** was originally based on the **NSCA HTTPd** server:

- It is the most popular **HTTP** server today.
- It is extensible, using modules.

As you can see in Figure 7.1, the **NetCraft January 2016** web server survey shows that **Apache** runs on nearly 40% of all active web servers. You can learn more about the survey on the **Internet**: <http://news.netcraft.com/archives/2016/01/26/january-2016-web-server-survey.html>



**Fig. 7.1: Web Server Developers: Market Share of Active Sites (Retrieved from [netcraft.com](http://news.netcraft.com/archives/2016/01/26/january-2016-web-server-survey.html))**

## 7.2.1 Install Apache

Apache is available within Ubuntu's default software repositories, so we will install it using conventional package management tools.

We will begin by updating the local package index to reflect the latest upstream changes. Afterwards, we can install the `apache2` package:

- `sudo apt-get update`
- `sudo apt-get install apache2`

After confirming the installation, `apt-get` will install Apache and all required dependencies.

## 7.2.2 Install Apache

Modify our firewall to allow outside access to the default web ports.

```
sudo ufw enable
```

We can list the `ufw` application profiles by typing:

- `sudo ufw app list`

You should get a listing of the application profiles:

```
vagrant@vagrant:~$ sudo ufw app list
Available applications:
Apache
Apache Full
Apache Secure
Dovecot IMAP
Dovecot POP3
Dovecot Secure IMAP
Dovecot Secure POP3
OpenSSH
Postfix
Postfix SMTPS
Postfix Submission
```

## 7.2.3 Install Apache

As you can see, there are three profiles available for Apache:

- **Apache:** This profile opens only port 80 (normal, unencrypted web traffic)
- **Apache Full:** This profile opens both port 80 (normal, unencrypted web traffic) and port 443 (TLS/SSL encrypted traffic)
- **Apache Secure:** This profile opens only port 443 (TLS/SSL encrypted traffic)

## 7.2.4 Install Apache

```
sudo ufw enable
```

```
vagrant@vagrant:~$ sudo ufw status  
Status: active
```

To	Action	From
--	-----	----
Apache Full	ALLOW	Anywhere
22	ALLOW	Anywhere
80	ALLOW	Anywhere
443	ALLOW	Anywhere
Apache Full (v6)	ALLOW	Anywhere (v6)
22 (v6)	ALLOW	Anywhere (v6)
80 (v6)	ALLOW	Anywhere (v6)
443 (v6)	ALLOW	Anywhere (v6)

## 7.2.4 Install Apache

```
sudo ufw enable
```

For our purposes, we will allow incoming traffic for the **Apache Full** profile by typing:

- `sudo ufw allow 'Apache Full'`

You can verify the change by typing:

- `sudo ufw status`



## 7.2.5 Install Apache

We can check with the `systemd` init system to make sure the service is running by typing:

- `sudo systemctl status apache2`

```
vagrant@vagrant:~$ sudo systemctl status apache2
```

```
• apache2.service - LSB: Apache2 web server
  Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
  Active: inactive (dead) since Tue 2018-02-27 19:19:48 UTC; 8min ago
  Docs: man:systemd-sysv-generator(8)
  Process: 13267 ExecStop=/etc/init.d/apache2 stop (code=exited, status=0/SUCCESS)
  Process: 16298 ExecReload=/etc/init.d/apache2 reload (code=exited, status=0/SUCCESS)
  Process: 1344 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SUCCESS)
```

```
Feb 26 06:25:02 vagrant systemd[1]: Reloading LSB: Apache2 web server.
Feb 26 06:25:02 vagrant apache2[16298]: * Reloading Apache httpd web server apache2
Feb 26 06:25:02 vagrant apache2[16298]: *
Feb 26 06:25:02 vagrant systemd[1]: Reloaded LSB: Apache2 web server.
Feb 27 19:19:47 vagrant systemd[1]: Stopping LSB: Apache2 web server...
Feb 27 19:19:47 vagrant apache2[13267]: * Stopping Apache httpd web server apache2
Feb 27 19:19:47 vagrant apache2[13267]: *
Feb 27 19:19:48 vagrant systemd[1]: Stopped LSB: Apache2 web server.
Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
```

## 7.2.6 Install Apache

```
sudo systemctl stop apache2
```

```
sudo systemctl start apache2
```

```
sudo systemctl restart apache2
```

If you are simply making configuration changes:

```
sudo systemctl reload apache2
```

By default, Apache is configured to start automatically when the server boots. If this is not what you want, you can disable default config:

```
sudo systemctl disable apache2
```

To re-enable the service to start up at boot, you can type:

```
sudo systemctl enable apache2
```

Apache should now start automatically when the server boots again.

## 7.3.1 Main Configuration Files

To access the main configuration file for **Apache**, which has a simple SGML-like format, use the following:

- **On CentOS:**
  - `/etc/httpd/conf/httpd.conf`
- **On OpenSUSE:**
  - `/etc/apache2/httpd.conf`
- **On Ubuntu:**
  - `/etc/apache2/apache2.conf`
  - which includes
    - `/etc/apache2/mods-enabled/`
    - `/etc/apache2/conf-enabled/`
    - `/etc/apache2/sites-enabled/`

```
# /etc/apache2/
# |-- apache2.conf
# |    |-- ports.conf
# |    |-- mods-enabled
# |    |-- |-- *.load
# |    |-- |-- *.conf
# |    |-- conf-enabled
# |    |-- |-- *.conf
# |    |-- sites-enabled
# |    |-- |-- *.conf
```

## 7.3.2 Main Configuration Files

Some of the more important configuration options are:

- The address and port to serve traffic from:
  - `Listen 12.34.56.78:80`
- Number of processes to start, maximum number of threads or forks (depending on the **MPM** module used):
  - `MPM Config`
  - `<IfModule prefork.c>`
  - `StartServers 8`
  - `MinSpareServers 5`
  - `MaxSpareServers 20`
  - `ServerLimit 256`
  - `MaxClients 256`
  - `MaxRequestsPerChild 4000`
  - `</IfModule>`
- The location on the filesystem where the served documents reside:
  - `DocumentRoot "/var/www/html"`

## 7.4 Include Configuration Files

To allow for modification and flexibility in the **apache** configuration file, you can include other files and directories. This allows you to avoid one large configuration file and is useful for servers with multiple sites. Many distributions use this feature to enable or disable web server configurations by installing or removing packages.

The **OpenSUSE** distribution also allows for easy creation of additional **include** files and directories. To learn more, check out the [/etc/sysconfig/apache2](#) file.

Some of the default **include** directories are:

**CentOS:**

`/etc/httpd/conf.d/*.conf`

**OpenSUSE:**

`/etc/apache2/conf.d/`

**Ubuntu:**

`/etc/apache2/conf-enabled/`

`/etc/apache2/sites-enabled/`

`/etc/apache2/sites-available/`

```
charset.conf@ --> /etc/apache2/conf-available/charset.c
javascript-common.conf@ --> /etc/apache2/conf-available/javascr
localized-error-pages.conf@ --> /etc/apache2/conf-available/localize
other-vhosts-access-log.conf@ --> /etc/apache2/conf-available/other-vho
security.conf@ --> /etc/apache2/conf-available/security
serve-cgi-bin.conf@ --> /etc/apache2/conf-available/serve-cg
```

## 7.5 Other files

Other important files include the document root, log file locations, and module locations (enabled in the configuration file).

The default document root is:

- **CentOS:**
  - `/var/www/html/`
- **OpenSUSE:**
  - `/srv/www/htdocs/`
- **Ubuntu:**
  - `/var/www/html/`

The default log file location is:

- **CentOS:**
  - `/var/log/httpd/`
- **OpenSUSE:**
  - `/var/log/apache2/`
- **Ubuntu:**
  - `/var/log/apache2/`

```
access.log
error.log
error.log.1
error.log.2.gz
nbox-ssl_access.log
nbox-ssl_error.log
nbox-ssl_error.log.1
nbox-ssl_error.log.2.gz
other_vhosts_access.log
```

To load a module, use the following syntax:

```
LoadModule alias_module modules/mod_alias.so
```

## Exercise 7.1: Install Apache and create a simple `index.html` file to serve

Include text to indicate this is the default server.

### Solution 7.1

1. Make sure **Apache** is installed:

- On **CentOS**:

```
# yum install httpd mod_ssl
```

- On **OpenSUSE**:

```
# zypper install apache2
```

- On **Ubuntu**:

```
# apt-get install apache2
```

2. Create an **index.html** file to serve with **Apache** in the default **DocumentRoot**.

The contents of the **index.html** file should be:

```
<html>
<head>
  <title>This is my file</title>
</head>
<body>
  <h1>This is my default file</h1>
</body>
</html>
```

The default DocumentRoot directories are:

- On **CentOS, Ubuntu:**

`/var/www/html/`

- On **OpenSUSE:**

`/srv/www/htdocs/`



3. Make sure **Apache** is enabled and started:

- On **systemd** distributions:

```
# systemctl enable httpd
# systemctl start httpd
or
# systemctl enable apache2
# systemctl start apache2
```

4. Verify the page you created is visible using a web browser.

```
$ firefox http://<YOUR_IP_ADDRESS>/index.html
or
$ w3m -dump http://<YOUR_IP_ADDRESS>/index.html
```

# 5 Labs in total in later classes

**You should now be able to**

**Install and configure an Apache web server supporting:**

- - **Default configuration.**
- - **IP-based virtual hosts.**
- - **Name-based virtual hosts.**
- - **Directory-level security.**
- - **SSL encryption for a host.**