

Operation Sheet 3.2 : Installing Apache Web Server in Linux Server

DR. PATRICK D. CERNA



2017

Installing Apache Web Server in Ubuntu Linux Server

Objectives: After the end of the activity, the students will be able to:

- Install and Configure Apache Web Server in Linux
- Create Website and Deploy it in Apache Web Server
- Test the configuration

Procedure:

Step 1: Open VM Ware Ubuntu Linux Server, and now Login on the shell on the server as user your created during installation. The username may differ if you have chosen a different name during setup.

```
Ubuntu 16.04 LTS server1 tty1
administrator:
Password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-21-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

5 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

administrator@server1:~$ _
```

Step 2: Install the Apache Web Server using the code below:

```
sudo apt-get update
```

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

Step 3: Adjust the Firewall

Before we can test Apache, we need to modify our firewall to allow outside access to the default web ports. Assuming that you followed the instructions in the prerequisites, you should have a UFW firewall configured to restrict access to your server.

sudo ufw app list

You should get a listing of the application profiles:

Output

Available applications:

Apache

Apache Full

Apache Secure

OpenSSH

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)

Step 4: Allow Apache Full to open port 80 and port 443

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

sudo ufw allow 'Apache Full'

Step 5: Verify the status of ufw

You can verify the change by typing:

sudo ufw status

You should see HTTP traffic allowed in the displayed output:

Output

Status: active

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
Apache Full	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
Apache Full (v6)	ALLOW	Anywhere (v6)

Step 6: Check if Apache Web Server is running

When you have your server's IP address or domain, enter it into your **browser's address bar (This is through your Windows 7 /10 Client)**

[http://server_domain_or_IP_or http://10.0.4.242](http://server_domain_or_IP_or_http://10.0.4.242)

You should see the default Ubuntu 16.04 Apache web page, which should look something like this:



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

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

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. Calling `/usr/bin/apache2` directly will not work with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to any file apart of those located in `/var/www`, `public_html` directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

Step 7: Go to the HTML Directory

Type the following command to go HTML Directory

```
Cd /var/www
```

Step 8: Create an HTML Page by typing the command:

```
sudo nano home.html
```

Once the editor is open type the following HTML code:

```
<html>  
<body>  
<h1>Welcome to my Apache Web Server</h1>  
<p>Amasedgenalehu </p>  
</body>  
</html>
```

Step 9: Test the HTML page

When you have your server's IP address or domain, enter it into your **browser's address bar** (This is through your Windows 7 /10 Client)

<http://10.0.4.242/home.html>

Step 10: Starting, Stopping, and Restarting Apache Web Server

sudo /etc/init.d/apache2 start

sudo /etc/init.d/apache2 stop

sudo /etc/init.d/apache2 restart