How to Install and Configure Apache Web Server on Ubuntu

What is Apache Web Server?

Apache or Apache HTTP server is a free and open source web server, developed and maintained by the Apache Software Foundation. Its popularity can be judged by the fact that around 46% of the websites worldwide are powered by Apache. Apache allows website developers to serve their content over the web. It serves as a delivery man by delivering files requested by users when they enter a domain name in their browser's address bar.

This tutorial is about installing and configuring Apache2 on your Ubuntu system. The commands and procedures mentioned in this article have been run on an Ubuntu 18.04 LTS system. Since we are using the Ubuntu command line, the Terminal, in this article; you can open it through the system Dash or the *Ctrl+Alt+T* shortcut.

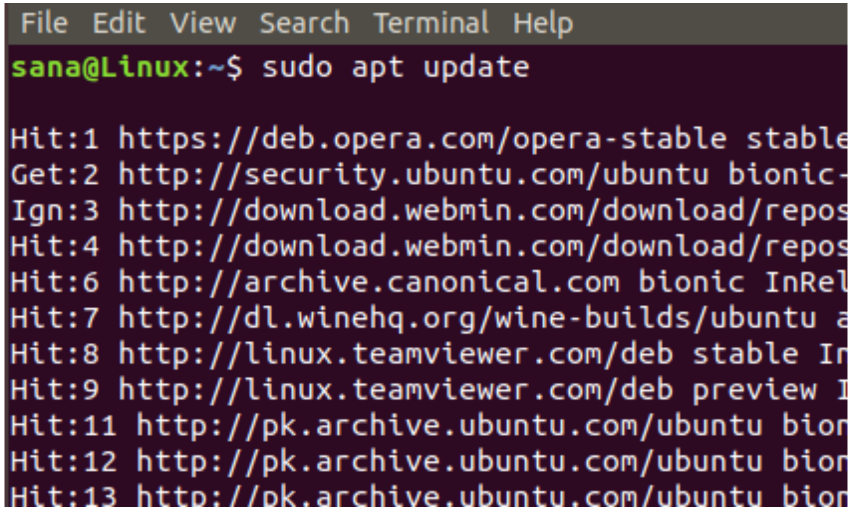
Install Apache 2 on Ubuntu Linux

Please follow the following steps in order to install the Apache2 software through Ubuntu official repositories.

Step 1: Update system repositories

You can download the latest version of a software by first updating the local package index of Ubuntu repositories. Open the Terminal and enter the following command in order to do so:

```
$ sudo apt update
```



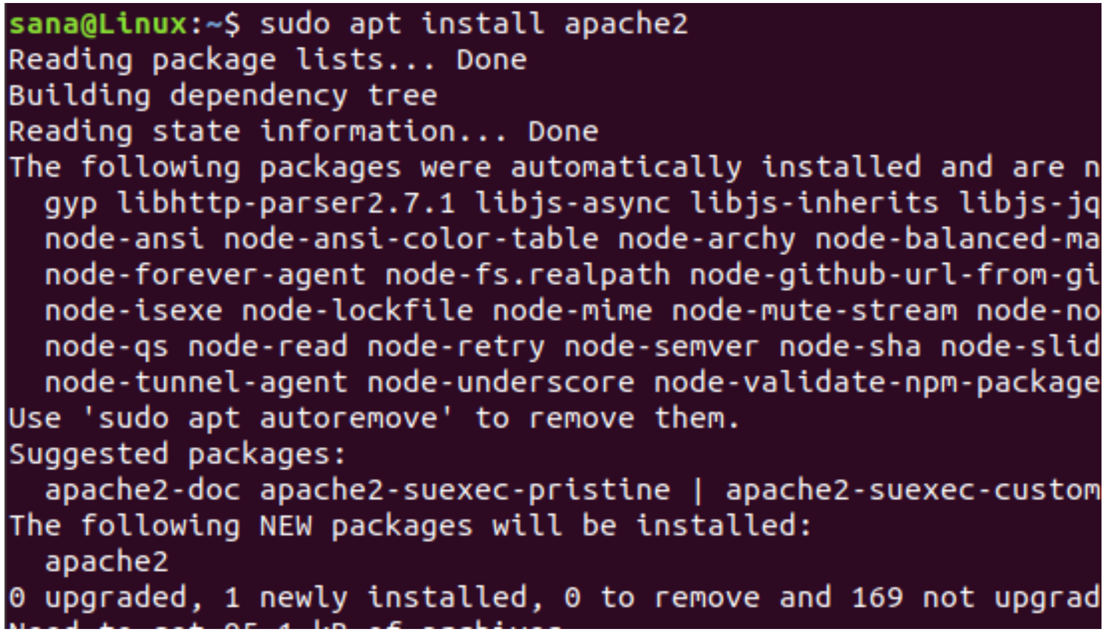
```
sana@Linux:~$ sudo apt update

Hit:1 https://deb.opera.com/opera-stable stable
Get:2 http://security.ubuntu.com/ubuntu bionic-
Ign:3 http://download.webmin.com/download/repos
Hit:4 http://download.webmin.com/download/repos
Hit:6 http://archive.canonical.com bionic InRel
Hit:7 http://dl.winehq.org/wine-builds/ubuntu a
Hit:8 http://linux.teamviewer.com/deb stable In
Hit:9 http://linux.teamviewer.com/deb preview I
Hit:11 http://pk.archive.ubuntu.com/ubuntu bion
Hit:12 http://pk.archive.ubuntu.com/ubuntu bion
Hit:13 http://pk.archive.ubuntu.com/ubuntu bion
```

Step 2: Install Apache 2 with the apt command

Next, enter the following command as sudo in order to install Apache2 and its required dependencies:

```
$ sudo apt install apache2
```



```
sana@Linux:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are n
  gyp libhttp-parser2.7.1 libjs-async libjs-inherits libjs-jq
  node-ansi node-ansi-color-table node-archy node-balanced-ma
  node-forever-agent node-fs.realpath node-github-url-from-gi
  node-isexe node-lockfile node-mime node-mute-stream node-no
  node-qs node-read node-retry node-semver node-sha node-slid
  node-tunnel-agent node-underscore node-validate-npm-package
Use 'sudo apt autoremove' to remove them.
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2
0 upgraded, 1 newly installed, 0 to remove and 169 not upgrad
```

You may be prompted with a y/n option to continue installation. Please enter Y, after which the installation procedure will begin.

Install NFS Server and Client on Ubuntu 18.04 LTS
posted on November 13, 2018 | under Linux, Ubuntu

6 Ways to Open Folders in Ubuntu 18.04 LTS
posted on February 14, 2019 | under Linux, Ubuntu

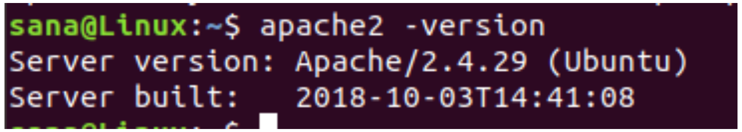
5 Ways to Check Available Memory in Ubuntu
posted on December 6, 2018 | under Linux, Ubuntu

How to Install VirtualBox on Ubuntu 18.04 LTS
posted on November 5, 2018 | under Desktop, Linux, Shell, Ubuntu

Step 3: Verify the Apache installation

When the installation is complete, you can check the version number and thus verify that Apache2 is indeed installed on your system by entering the following command:

```
$ apache2 -version
```



```
sana@Linux:~$ apache2 -version
Server version: Apache/2.4.29 (Ubuntu)
Server built:   2018-10-03T14:41:08
```

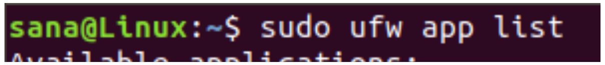
Configure the Firewall Settings

In order to configure Apache, we first need to allow outside access to certain web ports of our system and allow Apache on your UFW firewall.

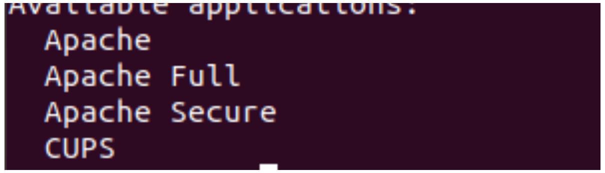
Step 1: List the UFW application profiles

In order to configure the firewall, let us first list the application profiles we will need to enable access to Apache. Use the following command to list such available applications:

```
$ sudo ufw app list
```



```
sana@Linux:~$ sudo ufw app list
```



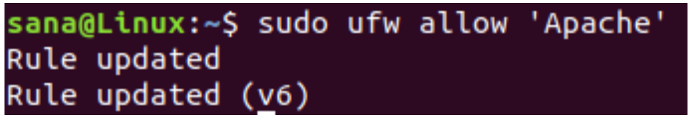
```
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
```

In the above output, you can see three Apache profiles all providing different levels of security; Apache being the one that provides maximum restriction with port 80 still open.

Step 2: Allow Apache on UFW and verify its status

Allowing Apache on UFW will open port 80 for network traffic, while providing maximum security to the server. Please configure UFW to allow Apache through the following command:

```
$ sudo ufw allow 'Apache'
```



```
sana@Linux:~$ sudo ufw allow 'Apache'
Rule updated
Rule updated (v6)
```

The status of UFW will now display Apache enabled on the firewall.

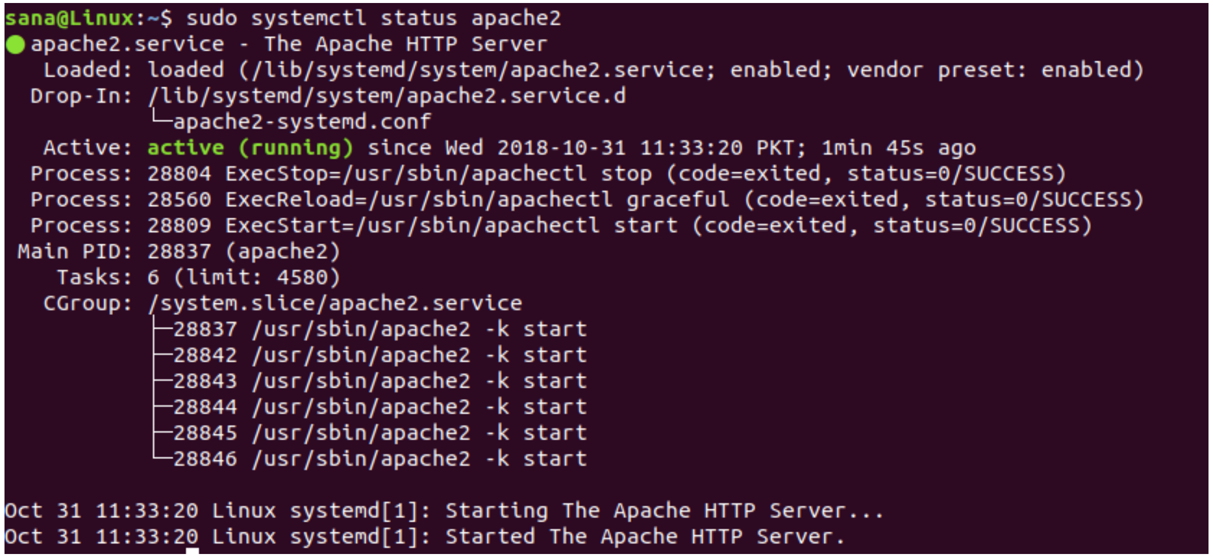
$ sudo ufw status

Configure the Apache Web server Settings

Step 1: Verify that the Apache service is running

The first step is to verify that the Apache2 service is up and running on your system, through the following command:

```
$ sudo systemctl status apache2
```



```
sana@Linux:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Wed 2018-10-31 11:33:20 PKT; 1min 45s ago
  Process: 28804 ExecStop=/usr/sbin/apachectl stop (code=exited, status=0/SUCCESS)
  Process: 28560 ExecReload=/usr/sbin/apachectl graceful (code=exited, status=0/SUCCESS)
  Process: 28809 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 28837 (apache2)
    Tasks: 6 (limit: 4580)
   CGroup: /system.slice/apache2.service
           ├─28837 /usr/sbin/apache2 -k start
           ├─28842 /usr/sbin/apache2 -k start
           ├─28843 /usr/sbin/apache2 -k start
           ├─28844 /usr/sbin/apache2 -k start
           ├─28845 /usr/sbin/apache2 -k start
           └─28846 /usr/sbin/apache2 -k start

Oct 31 11:33:20 Linux systemd[1]: Starting The Apache HTTP Server...
Oct 31 11:33:20 Linux systemd[1]: Started The Apache HTTP Server.
```

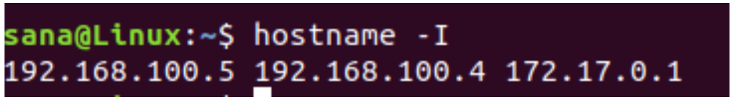
The status “active (running) verifies that the apache2 service is running.

Step 2: Verify that Apache is running properly and listens on your IP address

You can also verify if Apache is running by requesting a page from the Apache server. For this purpose, you can use your server's IP in order to access the Apache landing page.

Use the following command to know about your server's IP:

```
$ hostname -I
```

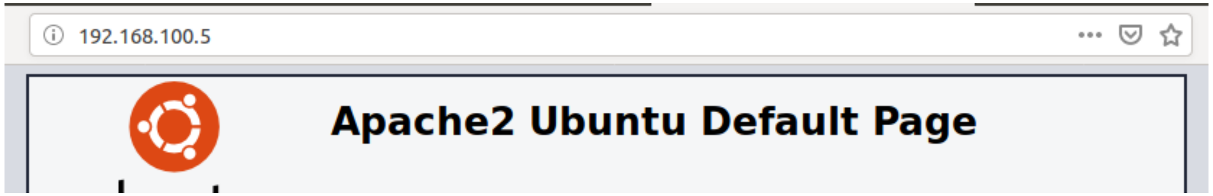


```
sana@Linux:~$ hostname -I
192.168.100.5 192.168.100.4 172.17.0.1
```

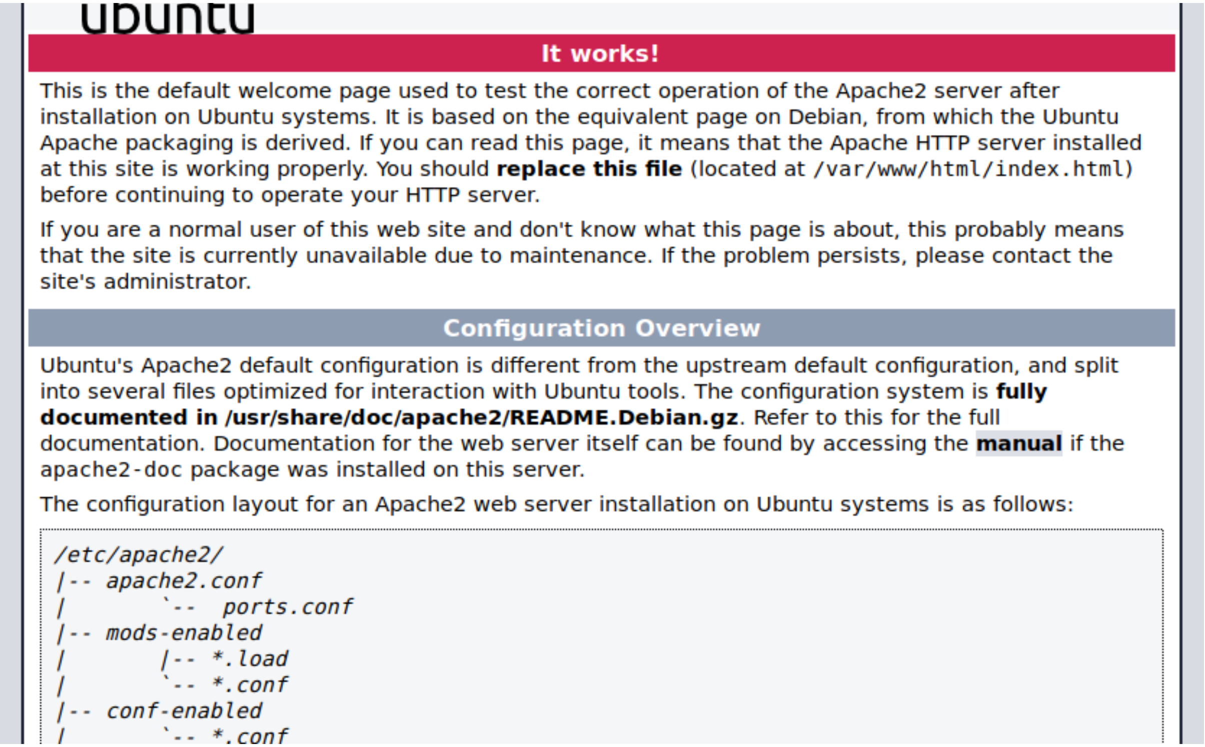
Then try the IPs, one by one from the output, in your web browser as follows:

http://server_IP

In my case, http://192.168.100.4 and http://192.168.100.5. Doing so will display the following Apache web page for Ubuntu, verifying that the Apache server is working properly.



Apache2 Ubuntu Default Page



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
```

Set Up Virtual Hosts in Apache

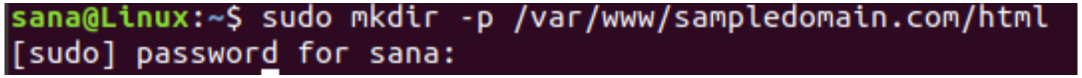
A virtual host is similar to what you have server blocks in Nginx. It is used to manage configurations for more than one domain from one server. We will present an example of how to set up a virtual host through the Apache server. We will set up a website named sampledomain.com by using the server block that is enabled by default in Apache for Ubuntu 18.

Step 1: Set up a domain name

The server block that is enabled by default is capable of serving documents from /var/www/html. However, we will create a directory at /var/www/ leaving the default directory intact.

Create this directory through the following command, replacing sampledomain.com by your respective domain name.

```
sudo mkdir -p /var/www/sampledomain.com/html
```

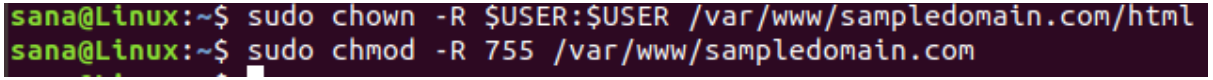


```
sana@Linux:~$ sudo mkdir -p /var/www/sampledomain.com/html
[sudo] password for sana:
```

Then assign the ownership of the directory through the following commands:

```
sudo chown -R $USER:$USER /var/www/sampledomain.com/html
```

```
sudo chmod -R 755 /var/www/sampledomain.com
```



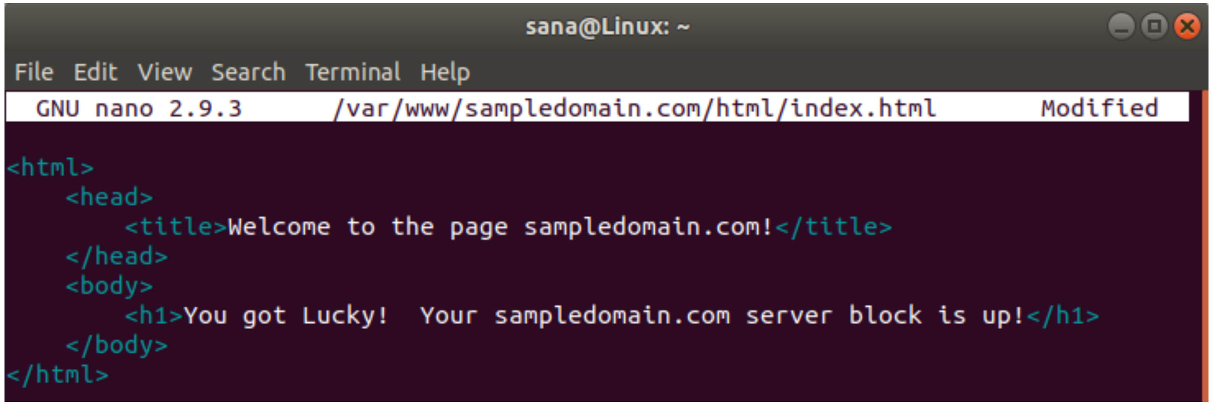
```
sana@Linux:~$ sudo chown -R $USER:$USER /var/www/sampledomain.com/html
sana@Linux:~$ sudo chmod -R 755 /var/www/sampledomain.com
```

Let us now create an index page that we can later access to test if Apache is running our domain name. Create an HTML file either through the Nano editor or any of your favorite text editor.

```
$ nano /var/www/sampledomain.com/html/index.html
```

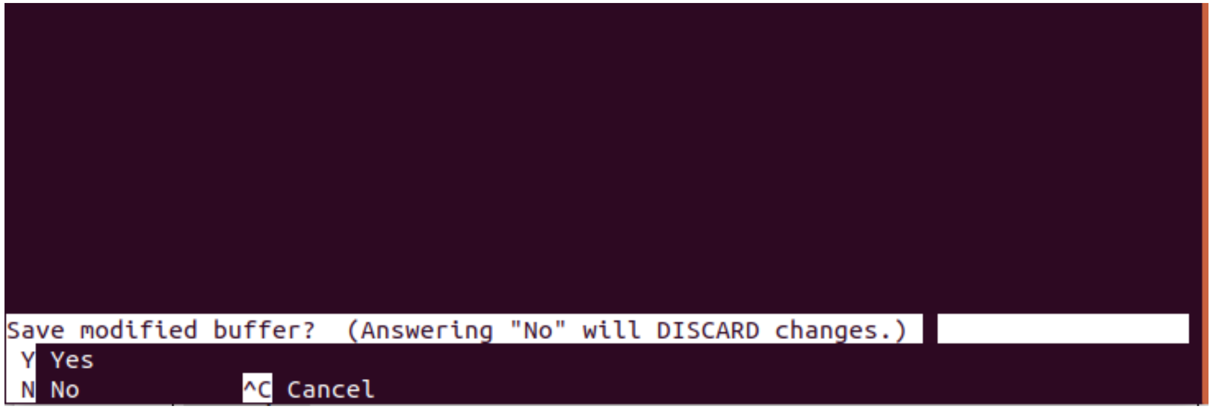
Enter the following HTML for the index page:

```
<html>
<head>
<title>Welcome to the page sampledomain.com!</title>
</head>
<body>
<h1>You got Lucky! Your sampledomain.com server block is up!</h1>
</body>
</html>
```



```
  GNU nano 2.9.3        /var/www/sampledomain.com/html/index.html        Modified

<html>
    <head>
        <title>Welcome to the page sampledomain.com!</title>
    </head>
    <body>
        <h1>You got Lucky!  Your sampledomain.com server block is up!</h1>
    </body>
</html>
```



```
Save modified buffer?  (Answering "No" will DISCARD changes.)
 Y Yes
 N No           ^C Cancel
```

We are using the nano editor to create the HTML file.

You can save a file in nano by using Ctrl+X and then enter Y and hitting Enter.

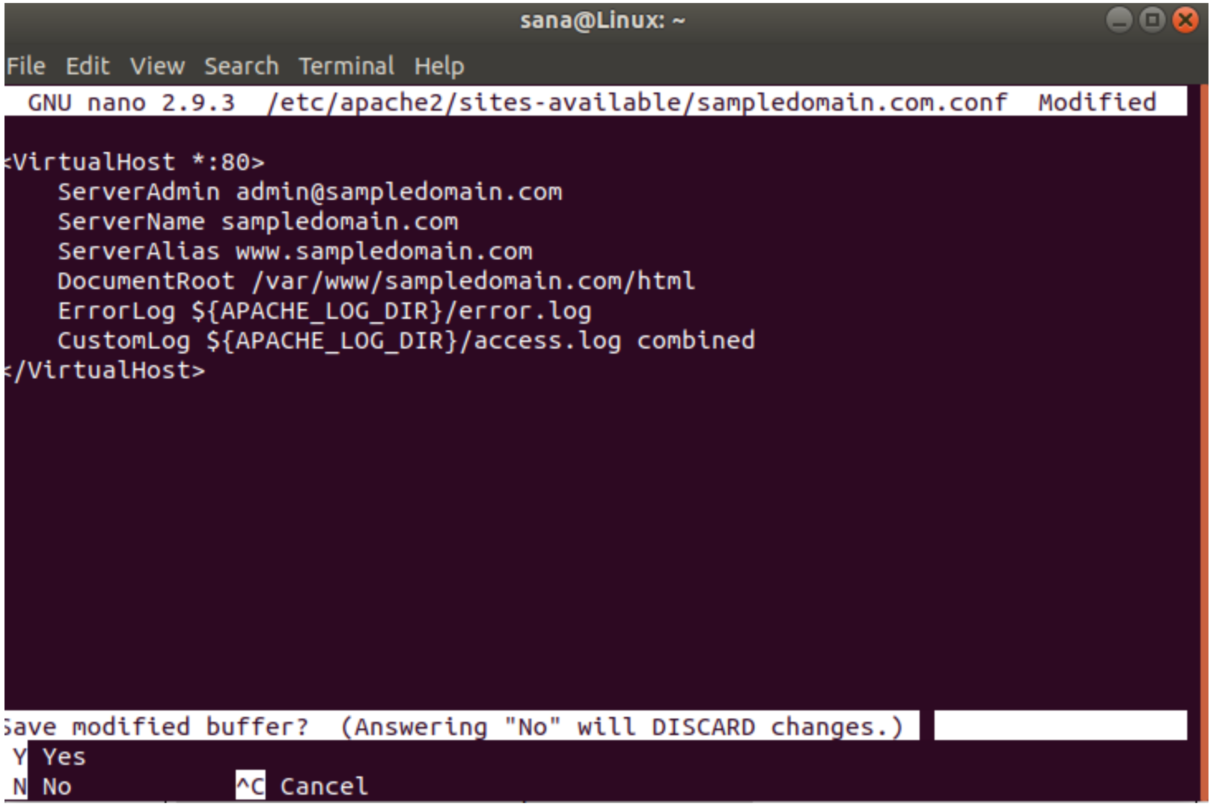
Apache needs a virtual host file to serve the contents of your server. The default configuration file for this purpose is already created but we will make a new one for our custom configurations.

```
$ sudo nano /etc/apache2/sites-available/sampledomain.com.conf
```

Enter the following customized configuration details for our domain name:

```
<VirtualHost *:80>
ServerAdmin admin@sampledomain.com
ServerName sampledomain.com
```

```
    ServerAlias www.sampledomain.com
    DocumentRoot /var/www/sampledomain.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    </VirtualHost>
```



```
  GNU nano 2.9.3  /etc/apache2/sites-available/sampledomain.com.conf  Modified

<VirtualHost *:80>
    ServerAdmin admin@sampledomain.com
    ServerName sampledomain.com
    ServerAlias www.sampledomain.com
    DocumentRoot /var/www/sampledomain.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

Save modified buffer?  (Answering "No" will DISCARD changes.)
 Y Yes
 N No          ^C Cancel
```

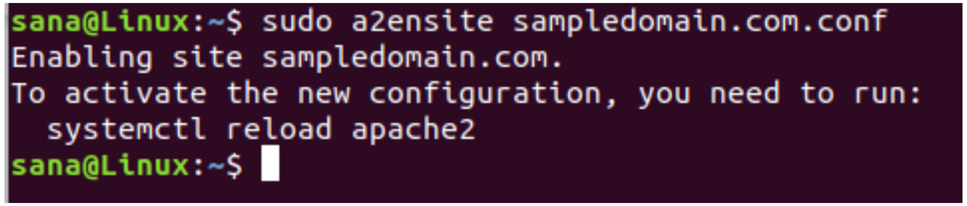
We are using the nano editor to create this .conf file.

You can save a file in nano by using *Ctrl+X* and then enter Y and hitting Enter.

Step 2: Enable the domain configuration file

Let us enable the configuration file we created with the a2ensite tool:

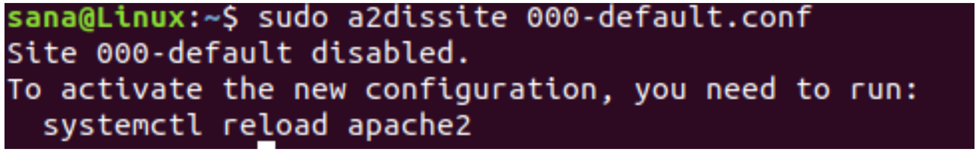
```
$ sudo a2ensite sampledomain.com.conf
```



```
sana@Linux:~$ sudo a2ensite sampledomain.com.conf
Enabling site sampledomain.com.
To activate the new configuration, you need to run:
  systemctl reload apache2
sana@Linux:~$
```

The output will suggest activating the new configuration but we can do it all collectively after running the following command that disables the original configuration file:

```
$ sudo a2dissite 000-default.conf
```



```
sana@Linux:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Now restart the Apache service:

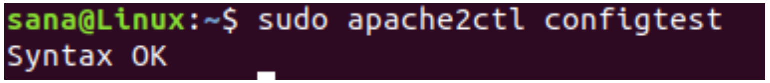
```
$ sudo systemctl restart apache2
```

Step 3: Test for errors

Finally, let us test if there are any configuration errors through the following command:

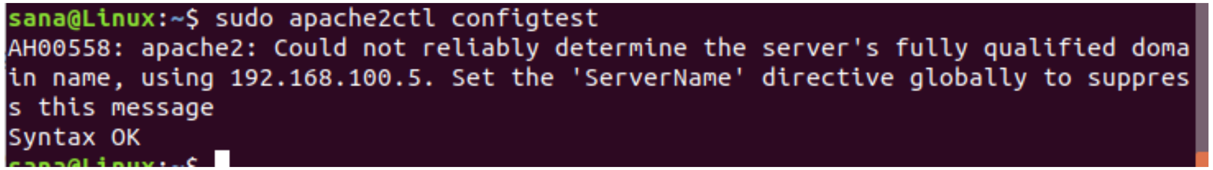
```
$ sudo apache2ctl configtest
```

If you do not get any errors, you will get the following output:



```
sana@Linux:~$ sudo apache2ctl configtest
Syntax OK
```

However, the following error is common in Ubuntu 18.04

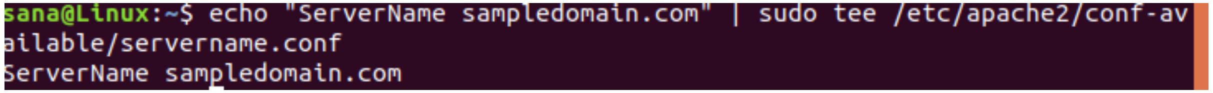


```
sana@Linux:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified doma
in name, using 192.168.100.5. Set the 'ServerName' directive globally to suppres
s this message
Syntax OK
sana@Linux:~$
```

Resolve the error:

Enter the following command in order to resolve the above-mentioned error:

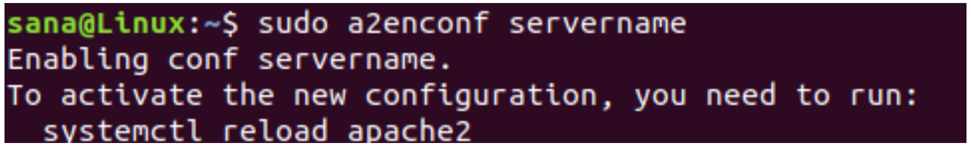
```
$ echo "ServerName sampledomain.com | sudo tee /etc/apache2/conf-
available/servername.conf
```



```
sana@Linux:~$ echo "ServerName sampledomain.com" | sudo tee /etc/apache2/conf-av
ailable/servername.conf
ServerName sampledomain.com
```

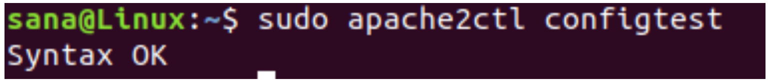
And then:

```
$ sudo a2enconf servername
```



```
sana@Linux:~$ sudo a2enconf servername
Enabling conf servername.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

Now when you check again for errors, you will see this error resolved through the following output:



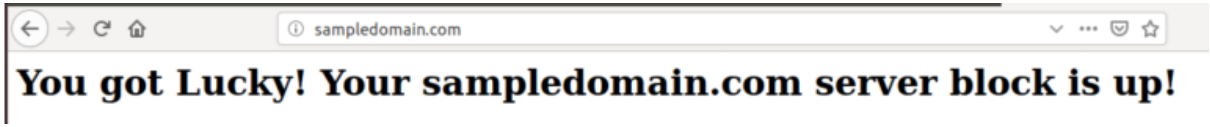
```
sana@Linux:~$ sudo apache2ctl configtest
Syntax OK
```

Step 4: Test if Apache is serving your domain name

Apache server is now configured to serve your domain name. This can be verified by entering your server name as follows in any of the web browsers running on your system:

http://sampledomain.com

The index page should display as follows, indicating that Apache is now ready to serve your server block!



You got Lucky! Your sampledomain.com server block is up!

Some Common Apache Management Commands

After setting up the web server, you might have to perform some basic management operations on Apache. Here are the commands that you can enter in your Terminal application for these operations.

```
sudo systemctl start apache2
```

Use this command as sudo in order to start the Apache server.

```
sudo systemctl stop apache2
```

Use this command as sudo in order to stop the Apache server when it is in start mode.

```
sudo systemctl restart apache2
```

Use this command as sudo in order to stop and then start the Apache service again.

```
sudo systemctl reload apache2
```

Use this command as sudo in order to apply the configuration changes without restarting the connection.

```
sudo systemctl enable apache2
```

Use this command as sudo in order to enable Apache to be started every time you boot your system.

```
sudo systemctl disable apache2
```

Use this command as sudo in order to disable if you have set up Apache to be started every time you boot your system.

Conclusion

Through this article, you have learned to install and configure the Apache web server on your Ubuntu system. This includes making some changes to your UFW firewall and then configuring your web server for your IP address. We also recommend you to set up a virtual host through Apache; this will give you a basis on how to use Apache to host your files on the Internet. The basic Apache management commands will also help you as a web administrator to manage your web server in an optimal manner.

📅 November 23, 2018 📁 Linux, Ubuntu

← How to Install the Skype alternative Ring on Ubuntu 18.04 LTS

How to Download YouTube Videos to Ubuntu 18.04 LTS →

Privacy Policy Imprint