Setup the Apache Web Server and host the runestonet1.com website.

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.

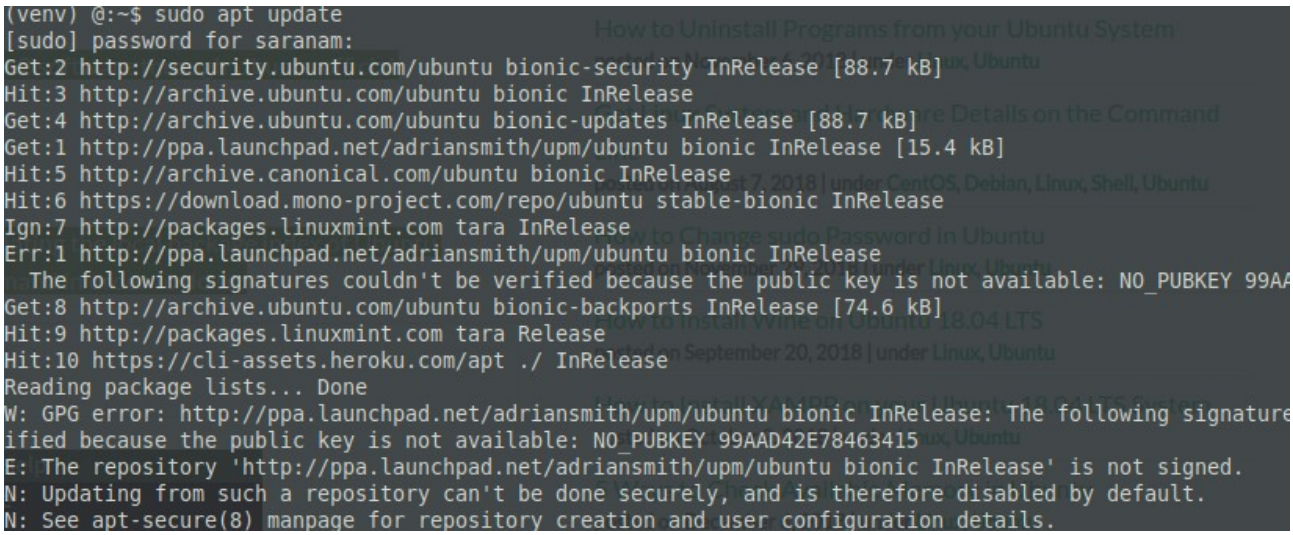
This tutorial is about installing and configuring Apache2 on your Ubuntu system. The commands and procedures mentioned in this article have been run on an Ubuntu 18.04 LTS system. Since we are using the Ubuntu command line, the Terminal, in this article;

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:



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:

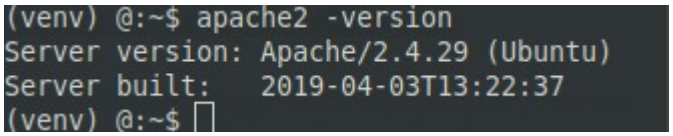
```
$ sudo apt install apache2
```

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

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
```



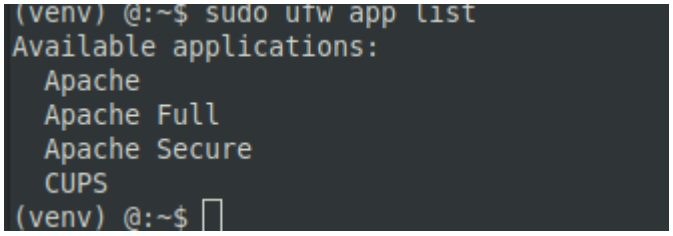
Configure the Firewall Settings

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:

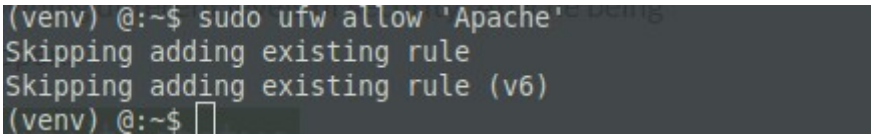
```
$ sudo ufw app list
```



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'
```



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

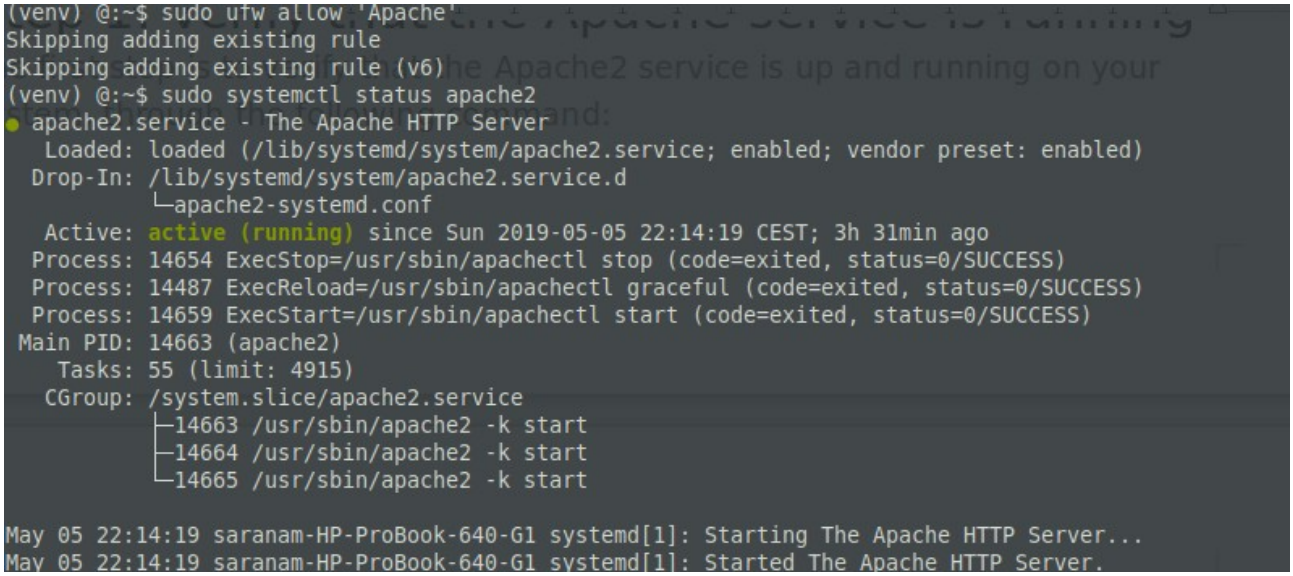
$ sudo ufw status

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
```



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

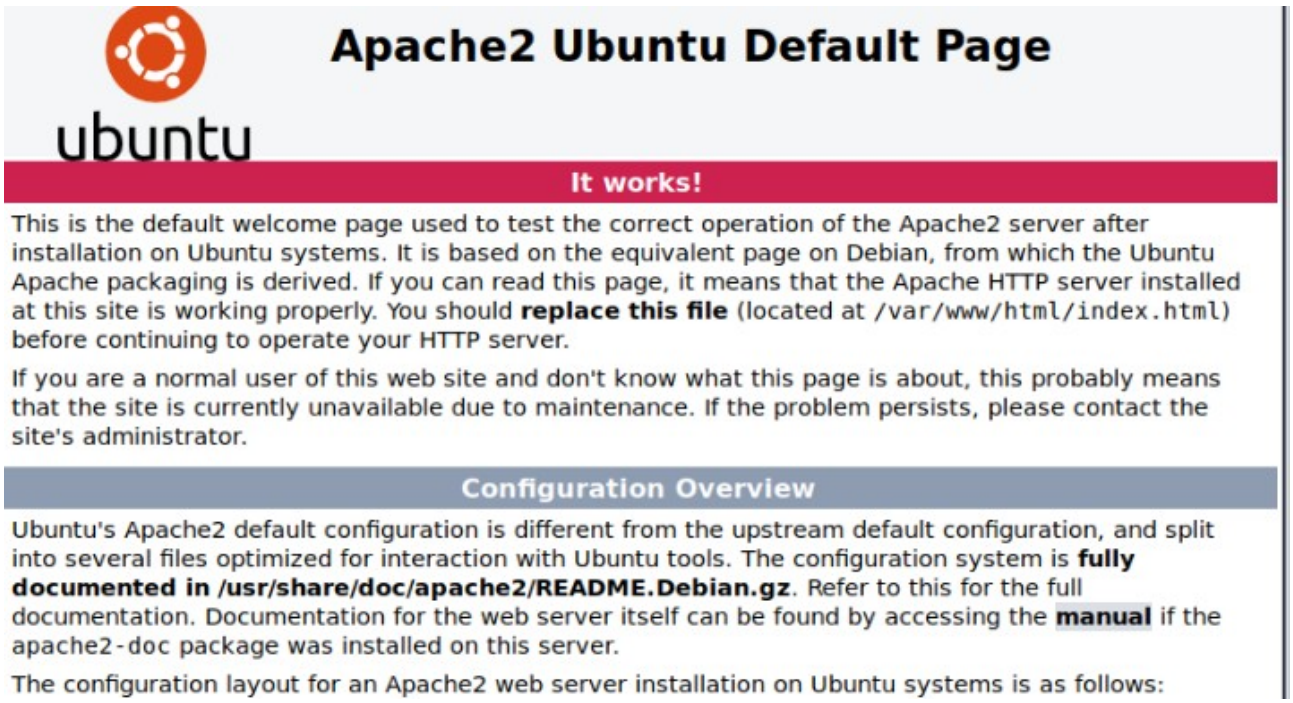
You can also verify if Apache is running by requesting a page from the Apache server. For this purpose, you can use your server’s IP in order to access the Apache landing page.

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

```
$ hostname -I
```

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

http://192.168.1.103 *or* ***http://localhost***



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 http://www.runestonet1.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 runestonet1.com by your respective domain name.

sudo mkdir -p /var/www/html/runestonet1.com/public_html

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.

sudo chown -R $USER:$USER /var/www/html/runestonet1.com/public_html

sudo vi /var/www/html/runestonet1.com/public_html/index.html

Enter the HTML:

```
<html>
 <head>
 <title>www.runestonet1.com</title>
 </head>
 <body>
 <h1>Hello, This is a test page for Runestone Team01 website</h1>
 </body>
</html>
```

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 vi /etc/apache2/sites-available/runestoneteam01.com.conf

```
<VirtualHost *:80>
    ServerAdmin admin@runestoneteam01.com
    ServerName runestoneteam01.com
    ServerAlias www.runestoneteam01.com
    DocumentRoot /var/www/runestoneteam01.com/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

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 runestoneteam01.com.conf
```

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
```

ow restart the Apache service:

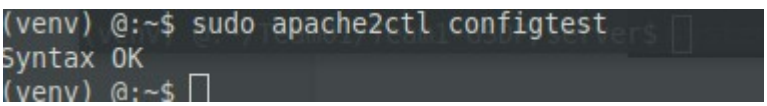
```
$ sudo systemctl restart apache2
```

Step 3: Test for errors

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

```
$ sudo apache2ctl configtest
```

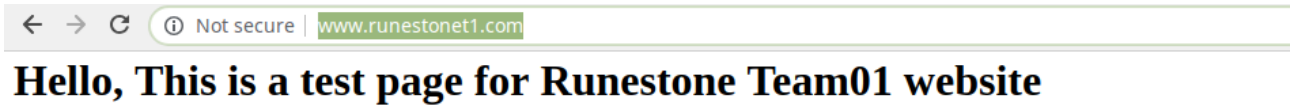
If you do not get any errors, you will get the following output:



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://www.runestonet1.com/



Hello, This is a test page for Runestone Team01 website

Reference:

https://www.digitalocean.com/community/tutorials/how-to-configure-the-apache-web-server-on-an-ubuntu-or-debian-vps