Comprehensive Guide to Installing Apache Web Server on Ubuntu 18.04 LTS.

On my journey towards DevOps and the Cloud, I will demonstrate in this article how to install the Apache web server on Ubuntu Linux, Customize HTML page and Automate Apache Server with Bash Script.

If you're reading this, you probably already know what Apache, Ubuntu, HTML and Bash Script, but let's quickly go over those together:

Apache: *Apache is the most widely used web server software. Developed and maintained by Apache Software Foundation, Apache is an open source software available for free. It runs on 67% of all web servers in the world. It is fast, reliable, and secure. It can be highly customized to meet the needs of many environments by using extensions and modules. Most WordPress hosting providers use Apache as their web server software. However, WordPress can run on other web server software as well.*

Ubuntu: Ubuntu is a Linux distro based on Debian. It is suitable for cloud computing, servers, desktops, and **internet of things (IoT)** devices. The main difference between Linux and Ubuntu is that the former is an operating system family based on Unix, while Ubuntu is a Linux distribution.

HTML: HTML stands for **HyperText Markup Language**. It is a standard markup language for web page creation. It allows the creation and structure of sections, paragraphs, and links using HTML elements (the building blocks of a web page) such as tags and attributes.

Bash Script: A bash script is a series of commands written in a file. These are read and executed by the bash program. The program executes line by line. For example, we can navigate to a certain path, create a folder and spawn a process inside it using the command line.

Before we get started, you'll need to do the following:

- Launch the command line terminal on your device and establish a connection to your cloud server by SSH.
- Have access to an account with sudo privileges.
- Lastly, we will need a server with Ubuntu 18.04 LTS installed.

SSH with the following command:

$ ssh user_name@host_ip_addres

- **ssh ("secure shell") command** instructs the system to establish an encrypted secure connection with the host machine.
- **User_name** represents the account that is being accessed on the host.
- **Host** refers to the machine, which can be a computer or a router that is being accessed. It can be an IP address (e.g., 102.468.2.28) or domain name.

Step 1: Update all packages on the server:

```
$  apt-get update -y
```

```
root@986c6b30fa3c:/home/cloud_user# apt-get update -y
```

Optional: Verify current Ubuntu version:

```
root@986c6b30fa3c:/home/cloud_user# cat /etc/os-release
NAME="Ubuntu"
VERSION="18.04.6 LTS (Bionic Beaver)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 18.04.6 LTS"
VERSION_ID="18.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=bionic
UBUNTU_CODENAME=bionic
```

Step 2: Install Apache:

```
$ apt-get install apache2 -y
$  apt-get upgrade apache2 -y
```

```
root@986c6b30fa3c:/home/cloud_user# apt-get install apache2 -y
root@986c6b30fa3c:/home/cloud_user# apt-get upgrade apache2 -y
```

If your server is protected by a firewall, and you need to open port HTTP (80) or HTTPS (443)

```
$  apt-get install firewalld
```

```
root@986c6b30fa3c:/home/cloud_user# apt-get install firewalld -y
```

After we have installed firewalld, we will enable the service and reboot our server. Keep in mind, enabling firewalld will cause the service to start up at boot.

Enable firewalld with the following command:

```
$ sudo systemctl enable firewalld
```

```
root@986c6b30fa3c:/home/cloud_user# systemctl enable firewalld
```

Follow this up with a quick server reboot.

Start firewall with the following command:

```
$ systemctl start firewalld
```

```
root@986c6b30fa3c:/home/cloud_user# systemctl start firewalld
```

Check status of firewall with the following command:

```
$ systemctl status firewalld
```

```
root@986c6b30fa3c:/home/cloud_user# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2022-11-18 23:24:34 UTC; 7min ago
     Docs: man:firewalld(1)
 Main PID: 21342 (firewalld)
    Tasks: 2 (limit: 2307)
   CGroup: /system.slice/firewalld.service
           └─21342 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid

Nov 18 23:24:33 986c6b30fa3c.mylabserver.com systemd[1]: Starting firewalld - dynamic firewa
Nov 18 23:24:34 986c6b30fa3c.mylabserver.com systemd[1]: Started firewalld - dynamic firewal
lines 1-11/11 (END)
```

Notice Active Status:

Enable firewall HTTP (port 80) and HTTPS (port 443) service with the following command:

```
$  sudo firewall-cmd --permanent --add-service=http
```

```
$  sudo firewall-cmd --permanent --add-service=https
```

```
root@986c6b30fa3c:/home/cloud_user# sudo firewall-cmd --permanent --add-service=http
success
root@986c6b30fa3c:/home/cloud_user# sudo firewall-cmd --permanent --add-service=https
success
```

Enable and start Apache service:

```
$  sudo systemctl start httpd
$  sudo systemctl enable httpd
$  sudo systemctl status httpd
```

```
root@986c6b30fa3c:/home/cloud_user# systemctl start apache2
root@986c6b30fa3c:/home/cloud_user# systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sy
sv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
root@986c6b30fa3c:/home/cloud_user# 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 Fri 2022-11-18 23:19:40 UTC; 1h 28min ago
 Main PID: 19190 (apache2)
    Tasks: 55 (limit: 2307)
   CGroup: /system.slice/apache2.service
           ├─19190 /usr/sbin/apache2 -k start
           ├─19192 /usr/sbin/apache2 -k start
           └─19193 /usr/sbin/apache2 -k start

Nov 18 23:19:40 986c6b30fa3c.mylabserver.com systemd[1]: Starting The Apache HTTP Server...
Nov 18 23:19:40 986c6b30fa3c.mylabserver.com systemd[1]: Started The Apache HTTP Server.
```

Step 5: Test the Web Page:

Let's test our web page by collecting the public IP address.

Obtain Public IP address with the following command:

```
$  curl -4 icanhazip.com
```

```
root@986c6b30fa3c:/home/cloud_user# curl -4 icanhazip.com
34.222.245.248
```

Once you've retrieved the public IP, type "http://your_public_ip" into your web browser and hit enter. You should see the following test page. If so, this means the server is properly configured.

Advance Part:

Step 6: Customize HTML Web Page.

Now go to cd /var/www/html/index.html modify the index.htm file with vi editor.

Enter the vi editor file

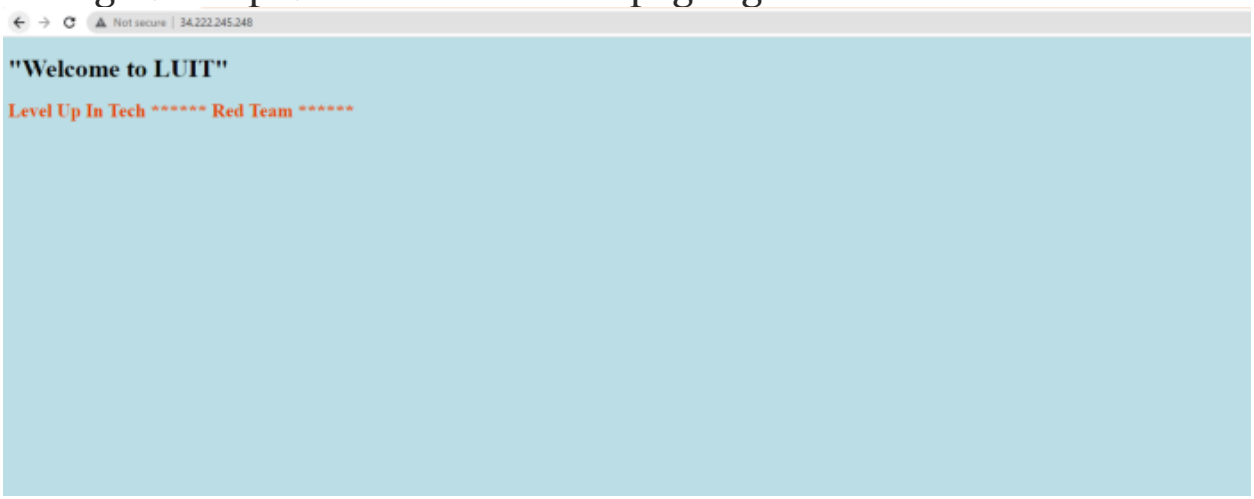
```
$ vi index.html
```

Edit the file and save with :wq.

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body style="background-color:powderblue;">

<h1>       "Welcome to LUIT"</h1>
<h2   style="color:Red ;">  Level Up In Tech ****** Red Team ******/h2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
:wq
```

Now grab the public IP and test the page again.



Congratulations to Customize the HTML file and running the server successfully.

Complex Part:

Step 6: Automate Apache Server with Bash Script.

To create our script, create a directory with bin then navigating to our /bin directory and creating our shell. Let's name it "luit.sh". Now we can make the file executable. The commands will look like this:

```
$ mkdir bin
$ cd bin
```

```
root@986c6b30fa3c:~# mkdir bin
root@986c6b30fa3c:~# ls
bin  snap
root@986c6b30fa3c:~# cd bin
root@986c6b30fa3c:~/bin# ls
```

Create .sh file for shell script and execute the .sh file

```
$ touch luit.sh
$ chmod +x luit.sh
```

```
root@986c6b30fa3c:~/bin# touch luit.sh
root@986c6b30fa3c:~/bin# chmod +x luit.sh
root@986c6b30fa3c:~/bin# ls
luit.sh
```

Now we can begin editing our script by typing the following:

```
$ sudo vi luit.sh
```

```
root@986c6b30fa3c:~/bin# sudo vi luit.sh
```

This will bring up the vi editor, where we can enter the contents of the script. Our script will begin with *"#!/bin/bash"* at the top, followed by:

```
!#/bin/bash

# This is Our First Project For LUIT Red Team.

sudo apt-get update -y

sudo apt-get upgrade -y

sudo apt install apache2 -y

sudo systemctl start apache2

sudo systemctl status apache2


echo "<body style="background-color:Green">

<h1>Welcome to LUIT</h1>
<h2>  Level Up In Tech ****** Red Team ****** </h2i>" | sudo tee /var/www/html/index.html
```

Note: we had to pipe into a *"tee"* command to get our text into the HTML file. This is required as our redirection is occurring in our shell, where there is no root access. Be sure to include this command to avoid permission errors.

Our script is complete and ready to run.

```
$ ./luit.sh
```

```
root@986c6b30fa3c:~/bin# sudo vi luit.sh
root@986c6b30fa3c:~/bin# ./luit.sh
./luit.sh: line 1: !#/bin/bash: No such file or directory
Hit:1 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:5 http://us-west-2.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [280
6 kB]
```

```
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4.25).
The following packages were automatically installed and are no longer required:
  linux-aws-5.4-headers-5.4.0-1081 linux-headers-5.4.0-1081-aws linux-image-5.4.0-1081-aws
  linux-modules-5.4.0-1081-aws
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
● 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 Fri 2022-11-18 23:19:40 UTC; 2h 26min ago
 Main PID: 19190 (apache2)
    Tasks: 55 (limit: 2307)
   CGroup: /system.slice/apache2.service
           ├─19190 /usr/sbin/apache2 -k start
           ├─19192 /usr/sbin/apache2 -k start
           └─19193 /usr/sbin/apache2 -k start

Nov 18 23:19:40 986c6b30fa3c.mylabserver.com systemd[1]: Starting The Apache HTTP Server...
Nov 18 23:19:40 986c6b30fa3c.mylabserver.com systemd[1]: Started The Apache HTTP Server.
<body style=background-color:Green>

<h1>Welcome to LUIT</h1>
<h2>  Level Up In Tech ****** Red Team ****** </h2i>
```

Now grab the public IP to verify the outcome.

******Congratulations! Mission Complete! ******