

STEP 5 — CREATING A VIRTUAL HOST FOR YOUR WEBSITE USING APACHE

In this project, you will set up a domain called projectlamp, but you can replace this with any domain of your choice.

Apache on Ubuntu 20.04 has one server block enabled by default that is configured to serve documents from the /var/www/html directory. We will leave this configuration as is and will add our own directory next next to the default one.

Create the directory for projectlamp using 'mkdir' command as follows:

```
sudo mkdir /var/www/projectlamp
```

Next, assign ownership of the directory with your current system user:

```
sudo chown -R $USER:$USER /var/www/projectlamp
```

Then, create and open a new configuration file in Apache's sites-available directory using your preferred command-line editor. Here, we'll be using vi or vim (They are the same by the way):

```
sudo vi /etc/apache2/sites-available/projectlamp.conf
```

This will create a new blank file. Paste in the following bare-bones configuration by hitting on i on the keyboard to enter the insert mode, and paste the text:

```
<VirtualHost *:80>
    ServerName projectlamp
    ServerAlias www.projectlamp
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/projectlamp
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

To save and close the file, simply follow the steps below:

- 1.Hit the esc button on the keyboard
- 2.Type :
- 3.Type wq. w for write and q for quit
- 4.Hit ENTER to save the file

You can use the ls command to show the new file in the sites-available directory

```
sudo ls /etc/apache2/sites-available
```

You will see something like this;

```
000-default.conf  default-ssl.conf  projectlamp.conf
```

With this VirtualHost configuration, we're telling Apache to serve projectlamp using /var/www/projectlamp/ as its web root directory. If you would like to test Apache without a domain name, you can remove or comment out the options ServerName and ServerAlias by adding a # character in the beginning of each option's lines. Adding the # character there will tell the program to skip processing the instructions on those lines.

You can now use a2ensite command to enable the new virtual host:

```
sudo a2ensite projectlamp
```

You might want to disable the default website that comes installed with Apache. This is required if you're not using a custom domain name, because in this case Apache's default configuration would overwrite your virtual host. To disable Apache's default website use a2dissite command, type:

```
sudo a2dissite 000-default
```

To make sure your configuration file doesn't contain syntax errors, run:

```
sudo apache2ctl configtest
```

Finally, reload Apache so these changes take effect:

```
sudo systemctl reload apache2
```

Your new website is now active, but the web root /var/www/projectlamp is still empty. Create an index.html file in that location so that we can test that the virtual host works as expected:

```
sudo echo 'Hello LAMP from hostname' $(curl -s http://169.254.169.254/latest/meta-data/public-hostname) 'with public IP'
$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4) > /var/www/projectlamp/index.html
```

Now go to your browser and try to open your website URL using IP address:

```
http://<Public-IP-Address>:80
```

If you see the text from 'echo' command you wrote to index.html file, then it means your Apache virtual host is working as expected. In the output you will see your server's public hostname (DNS name) and public IP address. You can also access your website in your browser by public DNS name, not only by IP – try it out, the result must be the same (port is optional)

```
http://<Public-DNS-Name>:80
```

You can leave this file in place as a temporary landing page for your application until you set up an index.php file to replace it. Once you do that, remember to remove or rename the index.html file from your document root, as it would take precedence over an index.php file by default.