

Install apache to your server

```
sudo apt install apache2 -y
```

To test type your IP in the browser if apache is successfully installed you will see apache2 default page If that step successfully completed then follow following steps to create Virtual hosts Step One — Create the Directory for your systems The first step that we are going to take is to make a directory structure that will hold the site data that we will be serving to visitors.

```
sudo mkdir -p /var/www/example.yourdomain.somaliren.org/public_html
```

```
sudo mkdir -p /var/www/test.yourdomain.somaliren.org/public_html
```

Step Two — Grant Permissions Give user and group to your directories

```
sudo chown -R $USER:$USER /var/www/example.yourdomain.somaliren.org/public_html
```

```
sudo chown -R $USER:$USER /var/www/test.yourdomain.somaliren.org/public_html
```

Set up proper permission

```
sudo chmod -R 755 /var/www
```

Step Three — Create Pages for Each Virtual Host

```
nano /var/www/example.yourdomain.somaliren.org/public_html/index.html
```

```
<html>
<head>
<title>Welcome to Example.yourdomain.somaliren.org!</title>
</head>
<body>
<h1>Success! The example.yourdomain.somaliren.org virtual host is working!</h1>
</body>
</html>
```

Save and close the file when you are finished.

```
nano /var/www/test.yourdomain.somaliren.org/public_html/index.html
```

```
<html>
<head>
<title>Welcome to Test.yourdomain.somaliren.org!</title>
</head>
<body> <h1>Success! The test.yourdomain.somaliren.org virtual host is working!</h1>
</body>
</html>
```

Step Four — Create New Virtual Host Files Virtual host files are the files that specify the actual configuration of our virtual hosts and dictate how the Apache web server will respond to various domain requests. Create the First Virtual Host File For simplifying you can copy the existing default configuration then customize

```
sudo nano /etc/apache2/sites-available/example.yourdomain.somaliren.org.conf
```

Put this text in the file

```
<VirtualHost *:80>
ServerAdmin admin@example.yourdomain.somaliren.org
ServerName example.yourdomain.somaliren.org
ServerAlias www.example.yourdomain.somaliren.org
DocumentRoot /var/www/example.yourdomain.somaliren.org/public_html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Save and close First Virtual host is Done Create another host just copy the previous configuration and customize

```
sudo cp /etc/apache2/sites-available/example.yourdomain.somaliren.org.conf
/etc/apache2/sites-available/test.yourdomain.somaliren.org.conf
```

```
sudo nano /etc/apache2/sites-available/test.yourdomain.somaliren.org.conf
```

Then make it look like this

```
<VirtualHost *:80>
ServerAdmin admin@test.yourdomain.somaliren.org
ServerName test.yourdomain.somaliren.org
ServerAlias www.test.yourdomain.somaliren.org
DocumentRoot /var/www/test.yourdomain.somaliren.org/public_html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Step Five — Enable the New Virtual Host Files We can use the [a2ensite](#) tool to enable each of our sites like this:

```
sudo a2ensite example.yourdomain.somaliren.org.conf
```

```
sudo a2ensite test.yourdomain.somaliren.org.conf
```

Step six--- disable the default site defined in 000-default.conf:

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

Finally restart apache service and test you pages

```
sudo systemctl restart apache2
```

Installing and enabling Certificate to your site Step1 Install SSL On Apache Server We will free and opensource certificate called letsencrypt To use that package you should install [python-certbot-apache](#)

```
sudo apt install python-certbot-apache
```

Step 2 Obtaining an SSL Certificate

```
sudo certbot --apache -d your_domain
```

f this is your first time running certbot, you will be prompted to enter an email address and agree to the terms of service. After doing so, certbot will communicate with the Let's Encrypt server, If that's successful, certbot will ask how you'd like to configure your HTTPS settings: The out will be like Output Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.

- 1: No redirect - Make no further changes to the webserver configuration.
- 2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or if you're confident your site works on HTTPS. You can undo this change by editing your web server's configuration.

Select the appropriate number [1-2] then [enter] (press 'c' to cancel): You can choose which option you want if you choose option 2 you will not be able to visit your site without https Select your choice then hit ENTER. The configuration will be updated, and Apache will reload to pick up the new settings. Certbot If that become successful you will get congratulation message