**Apache /mysql**

#update a list of packages in package manager

sudo apt update

#run apache2 package installation

sudo apt install apache2

To verify that apache2 is running as a Service in our OS, use following command

sudo systemctl status apache2

L

Another way to retrieve your Public IP address

curl -s http://<ip-address>

**MYSQL**

sudo apt install mysql-server

When the installation is finished, it’s recommended that you run a security script that comes pre-installed with MySQL. This script will remove some insecure default settings and lock down access to your database system. Start the interactive script by running:

sudo mysql\_secure\_installation

If you enabled password validation, you’ll be shown the password strength for the root password you just entered and your server will ask if you want to continue with that password. If you are happy with your current password, enter Y for “yes” at the prompt:

1FORyou.

Estimated strength of the password: 100

Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : y

For the rest of the questions, press Y and hit the ENTER key at each prompt. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes you have made.

When you’re finished, test if you’re able to log in to the MySQL console by typing:

sudo mysql

This will connect to the MySQL server as the administrative database user **root**, which is inferred by the use of sudo when running this command. You should see output like this:

# INSTALLING PHP

You have Apache installed to serve your content and MySQL installed to store and manage your data. [PHP](https://www.php.net/) is the component of our setup that will process code to display dynamic content to the end user. In addition to the php package, you’ll need php-mysql, a PHP module that allows PHP to communicate with MySQL-based databases. You’ll also need libapache2-mod-php to enable Apache to handle PHP files. Core PHP packages will automatically be installed as dependencies.

To install these 3 packages at once, run:

sudo apt install php libapache2-mod-php php-mysql

To know the php version we can use

php –v

# 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/edumix

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

sudo chown -R $USER:$USER /var/www/edumix

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:

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 edumix

ServerAlias [www.edumix](http://www.edumix)

ServerAdmin webmaster@localhost

DocumentRoot /var/www/edumix

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 edumix.conf

With this VirtualHost configuration, we’re telling Apache to serve edumix using **/var/www/**edumix 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 Edumix on 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/edumix/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.

# ENABLE PHP ON THE WEBSITE

With the default **DirectoryIndex** settings on Apache, a file named index.html will always take precedence over an index.php file. This is useful for setting up maintenance pages in PHP applications, by creating a temporary index.html file containing an informative message to visitors. Because this page will take precedence over the index.php page, it will then become the landing page for the application. Once maintenance is over, the index.html is renamed or removed from the document root, bringing back the regular application page.

In case you want to change this behavior, you’ll need to edit the **/etc/apache2/mods-enabled/dir.conf** file and change the order in which the **index.php** file is listed within the **DirectoryIndex** directive:

sudo vim /etc/apache2/mods-enabled/dir.conf

<IfModule mod\_dir.c>

#Change this:

#DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm

#To this:

DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm

</IfModule>

After saving and closing the file, you will need to reload Apache so the changes take effect:

sudo systemctl reload apache2